AFRL-HE-WP-TR-2004-0191

STINFO COPY

United States Air Force Research Laboratory

Agent-Based Modeling and Behavior Representation (AMBR)

Stephen Deutsch
Richard W. Pew
Yvette J. Tenney
David E. Diller
Katherine Godfrey
Sandra Spector
Brett Benyo
Sachin Date

BBN Technologies
10 Moulton Street
Cambridge, MA 02138

Kevin A. Gluck

Air Force Research Laboratory

September 2004

Final Report for the Period June 1999 to September 2004

20050425 068

Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2004-0191

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

//Signed//

MARIS M. VIKMANIS
Chief, Warfighter Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
September 2004

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
June 1999 - September 2004

4. TITLE AND SUBTITLE
Agent-Based Modeling and Behavior Representation (AMBR)

5a. CONTRACT NUMBER
F33615-99-C-6002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63231F

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
49230401

6. AUTHOR(S)
Stephen Deutsch, Richard W. Pew, Yvette J. Tenney, David E. Diller, Katherine Godfrey, Sandra Spector, Brett Benyo, Sachin Date, Kevin A. Gluck

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
BBN Technologies
10 Moulton Street
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER
BBN Report Number 8404

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Cognitive Systems Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-WP-TR-2004-0191

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report documents accomplishments and lessons learned in a multi-year project to examine the ability of a range of integrative cognitive modeling architectures to predict human behavior in a common task environment. This Agent-Based Modeling and Behavior Representation (AMBR) project involved a series of human performance model evaluations in which the behavior of computer models could be compared to each other and to actual human operators performing the identical tasks. The first comparison challenged the modelers to build dynamically realistic human cognitive models of multiple task management and attention sharing, by simulating the behavior of an air traffic controller (ATC) operating in a simplified ATC task. The second comparison challenged the modelers to build computational process models that simulated the learning of new concepts in the context of executing the task and to make a priori predictions of human behavior in a transfer condition. This report consists of chapters of a forthcoming book, plus appendices detailing the AMBR methodology.

15. SUBJECT TERMS Cognitive Modeling, Behavioral Representation, Human Cognitive Models, Air Traffic Simulation, Human Learning Models, AMBR, Human Performance Modeling, Attention Sharing, Multiple Task Management

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFED	SAR	278	19b. TELEPHONE NUMBER *(include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

THIS PAGE LEFT INTENTIONALLY BLANK

Contents

List of Figures

List of Tables

Preface

This report is intended to meet the requirements for a Final Report ("Scientific and Technical Report," Item Number A001) under contract F33615-99-C-6002 with the U.S. Air Force Research Laboratory. This report consists of Chapters 1, 2, 3, 8, and 11 of the final draft of the forthcoming book by Gluck and Pew (in press): *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation.* These chapters provide the background for the project, describe in detail the work accomplished in this and related phases of the contract together with the summary of lessons learned and recommendations for further research. (The chapters have been edited slightly for this report so that they stand alone and provide explicit references to other parts of the book.) The appendices of this report contain the experimental materials used in Experiment 2.

Acknowledgements

The design of Experiment 2, which focused on concept learning, required the collection of significantly more human subject data than originally planned at the beginning of the project. As noted above, Harold Hawkins and ONR stepped up with funds to support this new requirement. Amy Bolton and Gwen Campbell were instrumental in making this possible, but it was actually Randolph Astwood, Jr., of Jardon Howard Technologies, and David Holness, of the NAVAIR Orlando Training Systems Division, who collected the data. We appreciate their involvement.

The expert panel for the second comparison, which focused on models of concept learning, consisted of Gwen Campbell, NAVAIR Orlando Training Systems Division, Harold Hawkins, Office of Naval Research, Bonnie John, Carnegie Mellon University and Bradley Love, University of Texas at Austin. These people gave generously of their time and effort and it is unquestionably the case that the AMBR Model Comparison benefited from their participation.

THIS PAGE LEFT INTENTIONALLY BLANK

1. Background, Structure, and Preview of the Model Comparison

Kevin A. Gluck, Richard W. Pew, Michael J. Young

1.1 Introduction

The U.S. military services have developed a variety of systems that allow for synthetic human behavior representation (HBR) in virtual and constructive simulation. Examples include the Army's Modular Semi-Automated Forces (ModSAF), the Navy and Marine Corps' Small Unit Tactical Trainer (SUTT), the Air Force's Advanced Air-to-Air System Performance Evaluation Model (AASPEM), and the Joint Services' Command Forces (CFOR) project. Pew and Mavor (1998) describe these systems and others, then note that although it is possible to represent human behavior in these systems, the state of the human representation is almost always rudimentary. In the words of Pew and Mavor:

> This lack of human performance representation in models becomes more significant as the size, scope, and duration of wargaming simulations continues to grow. In the future, these limitations will become more noticeable as greater reliance is placed on the outcomes of models/simulations to support training and unit readiness, assessments of system performance, and key development and acquisition decisions. (p. 44)

To begin addressing the problems associated with limited HBR capability, developers of these and future military modeling and simulation systems should begin to draw more from cognitive, social, and organizational theory. In particular, Pew and Mavor (1998) suggest that these modeling systems would benefit from a closer association with the developers of integrative HBR architectures.

In the psychology literature, the term *architecture* is often used instead of *system* (e.g., Anderson, 1983, 1993; Newell, 1990; Pylyshyn, 1991). A psychological architecture differs from other modeling and simulation systems in that it makes *a priori* assumptions that constrain the representations and processes available for use in a model on the basis of the theories underlying the architecture. By virtue of these constraints, architectures are a distinct subset of the total set of possible human representation systems. Chapter 3 of the Pew and Mavor (1998) text provides a description of the major characteristics of 11 integrative architectures. Ritter, Shadbolt, Elliman, Young, Gobet, and Baxter (2003) describe seven more. Morrison (2004) reviewed the characteristics of many of the same architectures, and added still another six to the list. That is a total of at least 24 human representation architectures included in recent reviews.

We recently went through that list of two dozen architectures to confirm the availability of each as implemented in software that can be used to develop models and that perhaps also could be integrated into a larger simulation system. The subset of psychologically-inspired human representation architectures that meet this availability criterion[1] is listed in Table 1. We will not review the characteristics of these architectures here, because that would be redundant with the three recent reviews, but we encourage the interested reader to seek out the references above or read about the architectures on their respective websites.

Table 1: Human Behavior Representation Architectures Available for Use

Architecture	***For additional information:***
ACT-R	http://act-r.psy.cmu.edu/
APEX	http://www.andrew.cmu.edu/~bj07/apex/
ART	http://web.umr.edu/~tauritzd/art/
Brahms	http://www.agentisolutions.com/home.htm
CHREST	http://www.psyc.nott.ac.uk/research/credit/projects/CHREST
C/I	http://www.inst.msstate.edu/SAL/adapt.html
Clarion	http://www.cogsci.rpi.edu/~rsun/clarion.html
CogAff	http://www.cs.bham.ac.uk/~axs/cogaff.html
Cogent	http://cogent.psyc.bbk.ac.uk
COGNET/iGEN	http://www.chiinc.com/
D-OMAR	http://omar.bbn.com/
EPAM	http://www.pahomeschoolers.com/epam/
EPIC	http://www.umich.edu/~bcalab/epic.html
MicroPsi	http://www.informatik.hu-berlin.de/~bach/artificial-emotion/
Micro Saint, HOS, IPME	http://www.maad.com/MaadWeb/products/prodma.htm
MIDAS	http://caffeine.arc.nasa.gov/midas/
PDP++	http://psych.colorado.edu/~oreilly/PDP++/PDP++.html
SAMPLE[2]	http://www.cra.com/sample
Soar	http://www.soartechnology.com

[1] Absent from this list are Sparse Distributed Memory (SDM; Kanerva, 1993), Contextual Control Model (CoCoM; Hollnagel, 1993), and 4CAPS (Just, Carpenter, & Varma, 1999). SDM does not exist in simulation form (Kanerva, personal communication, October 10, 2003). CoCoM does not exist as executable code in the public domain (Hollnagel, personal communication, October 13, 2003). 4CAPS exists in simulation form, but is not being released publicly until adequate documentation and pedagogical materials are in place (Varma, personal communication, February 26, 2004).

[2] Contact Karen Harper (kharper@cra.com) directly to obtain the SAMPLE software.

The existence of such an assortment of HBR architectures is an indication of the health and vitality of this research area. Yet there is considerable room for improvement. All of the architectures have shortcomings in their modeling capabilities and none of them are as easy to use as we would like them to be. There is enormous interest in greater breadth, increased predictive accuracy, and improved usability in models of human performance and learning. These interests motivated the creation of a research project that would move the field in those directions.

1.2 The AMBR Model Comparison

This unique project, called the Agent-based Modeling and Behavior Representation (AMBR) Model Comparison, was sponsored primarily by the US Air Force Research Laboratory (AFRL), with additional funding from the Office of Naval Research (ONR). The AMBR Model Comparison involved a series of human performance model evaluations in which the behaviors of computer models were compared to each other and to the behaviors of actual human operators performing the identical tasks.

1.2.1 The Approach

Considered in isolation, there is nothing unique about developing models and comparing them to human data. Cognitive science and other related disciplines are replete with such activities. The unique nature of the project is revealed only through consideration of the details of our approach and how it relates to similar efforts.

A previous research project with which the AMBR Comparison shares a close affinity is the Hybrid Architectures for Learning Project sponsored by ONR in the mid- to late-1990's. "Hybrid Architectures" was committed to improving our understanding of human learning by funding the development of various cognitive architecture-based and machine learning-based models in three different learning contexts. The modeling goal was " . . . to run the basic hybrid model on a selected task to verify the model's performance relative to the actual human data and to evolve the model, increasing the match between the learned performances, to obtain a better predictive/explanatory model of the human process" (Gigley & Chipman, 1999, p. 2). The emphases on (a) iterative improvements to computational models and model architectures and on (b) evaluating these improvements through comparison to human data both find parallel emphases in AMBR. There was even an intention in Hybrid Architectures to eventually conduct a thorough comparison of the models that had been developed for the various tasks, but

unfortunately the funding for the project disappeared before a final comparison took place. The major methodological differences between the two projects are that (a) all of the AMBR modelers developed models of the same tasks, in order to facilitate comparison, and (b) detailed comparison of the models was an integral part of AMBR and took place on a recurring basis starting early on.

Another effort which can help illuminate some of the distinctive characteristics of the AMBR Model Comparison is the comparison of models of working memory that took place in the late 1990's. The working memory model comparison initially took the form of a symposium and eventually evolved into a book on the topic (Miyake & Shah, 1999). Their goal was to compare and contrast existing models of working memory by having each modeler address the same set of theoretical questions about their respective model's implementation. There are probably more differences than similarities between their effort and AMBR, although both approaches were effective in achieving their objectives. One distinction is that the AMBR models were all implemented in computational process models that can interact with simulated task environments, whereas the working memory models came from an assortment of modeling approaches, including verbal/conceptual theories. Another distinction is that the AMBR Model Comparison was partially motivated by an interest in encouraging computational modelers to improve the implementations and/or applications of their architectures by pushing on their limits in new ways, whereas the working memory model effort did not fund the development of new models or architectural changes. A third distinction is that, as mentioned previously, all of the AMBR modelers were required to address the same task scenarios, whereas the working memory modelers each focused on a task of their own choosing. In Chapter 12 of the Miyake and Shah (1999) book, Kintsch, Healy, Hegarty, Pennington, and Salthouse (1999) applaud the success of the editors' "common questions" approach to comparing the models. It is noteworthy that they then go on to recommend the following for model comparisons:

> . . . we would like to emphasize that, to the extent that direct experimental face-offs among models are possible, they should certainly be encouraged. Obviously, such comparisons would be very informative, and much more could be and should be done in this respect than has heretofore been attempted. (p. 436)

Although not originally inspired by this quote, the strategy adopted in AMBR of having each model address the same experiment scenarios is consistent with the Kintsch et al. recommendation. It also is consistent with the proposal a decade earlier by Young, Barnard,

Simon, and Whittington (1989) that HCI researchers adopt the use of scenarios as a methodological route to models of broader scope.

Hopefully the previous paragraphs gave the reader an appreciation for the general research approach selected for the AMBR Model Comparison, but this tells us little of the precise process that was followed. There were two experiments in the AMBR Model Comparison, pursued sequentially. The first focused on multi-tasking and the second focused on category learning. Each of the two experiments involved the following steps:

(1) Identify the modeling goals – what cognitive/behavioral capabilities should be stressed?
(2) Select a task domain that requires the capabilities identified in (1) and that is of relevance to AF modeling and simulation needs.
(3) Borrow/Modify/Create a simulation of the task domain which either a human-in-the-loop or a human performance model can operate.
(4) Hold a workshop at which the model developers learn about the task and modeling environment and exchange ideas with the moderator concerning potential parameters that can be measured and constraints of the individual models that will need to be accommodated.
(5) Moderator team collects and disseminates human performance data.
(6) Modeling teams develop models that attempt to replicate human performance when performing the task.
(7) Expert panel convenes with the entire team to compare and contrast the models that were developed and the underlying architectures that support them.
(8) Share the results and lessons learned with the scientific community, to include making available the simulation of the task domain and the human performance data.

We should note that some of the data were withheld from the modelers in the second comparison, which focused on category learning. We'll say more about that below.

1.2.2 Manager, Moderator, and Modelers

The project involved people from a variety of organizations, representing government, industry, and academia. The Air Force Research Laboratory's Warfighter Training Research Division managed the effort. BBN Technologies served in the role of Model Comparison moderator. They designed the experiments, provided the simplified Air Traffic Control (ATC) simulation environment implemented in D-OMAR (Deutsch & Benyo, 2001; Deutsch, MacMillan, & Cramer, 1993), and collected data on human operators performing the task. Additional data for the second comparison (category learning) were collected at the University of Central Florida,

with supervision from colleagues at NAVAIR Orlando (Gwen Campbell and Amy Bolton). There were four modeling teams. Two of the teams (CHI Systems and a team from George Mason University and Soar Technology) were selected as part of the competitive bidding process at the beginning of the first comparison. A team from Carnegie Mellon University joined the first comparison in mid-course, with funding from the Office of Naval Research. Finally, a fourth modeling team, this one from the Air Force Research Laboratory's Logistics and Sustainment Division, participated on their own internal funding.

1.2.3 Goals

There were three goals motivating the AMBR Model Comparison, all of which bear a striking resemblance to the recommendations made by the National Research Council (NRC) Panel on Modeling Human Behavior and Command Decision Making (Pew & Mavor, 1998).

Goal 1: Advance the State of the Art. The first goal was to advance the state of the art in cognitive modeling. This goal is consistent with the spirit of the entire set of recommendations from the NRC panel, since their recommendations were explicitly intended as a roadmap for improving human and organizational behavior modeling. The model comparison process devised for this project provides a motivation and opportunity for human modelers to extend and test their architectures in new ways. As should be apparent in the subsequent sections of this report (see also Gluck and Pew, in press), there is ample evidence that these modeling architectures were challenged and improved as a direct result of their participation in this project.

Goal 2: Develop Mission-Relevant HBR Models. The second goal was to develop HBR models that are relevant to the Department of Defense mission, and therefore provide possible transition opportunities. This is consistent with the NRC panel recommendation to support model development in focused areas of interest to the DoD. The two modeling focus areas selected for AMBR were multi-tasking and category learning. We'll say more about each of those areas shortly.

Goal 3: Make Tasks, Models, and Data Available. The third goal was to make all of the research tasks, human behavior models, and human process and outcome data available to the public. This is consistent with the NRC panel recommendation for increased collection and dissemination of human performance data. We have described various subsets of the results from the AMBR Model Comparison at several different conferences over the last three years, resulting in almost three dozen conference papers and technical reports. This book, however, is the most

comprehensive source of information regarding the scientific output of the AMBR Model Comparison.

1.2.4 Experiment I: Multi-Tasking

The AMBR Model Comparison was divided into two experiments, with a different modeling focus in each. The modeling focus for Experiment 1 was multiple task management, because this area represents a capability that is not widely available in existing models or modeling architectures, and because more knowledge regarding how to represent this capability provides an opportunity to improve the fidelity of future computer-generated forces (CGF's). It was the responsibility of the Moderator (BBN) to select a task for simulation that emphasized multiple task management.

Two approaches, representing ends of a continuum of intermediate possibilities, were considered. One approach is to select a high-fidelity task that is of direct operational relevance, realistic complexity, and requires highly trained operators to be the participants. Alternatively, the task could be highly abstracted, almost like a video game that anyone could be expected to learn, but that captures the task management requirements of interest.

Clearly the high-fidelity approach would have greater practical significance and be more challenging from a modeling perspective. However, it would require extensive knowledge acquisition on the part of each modeling team, an investment that would detract from the time and effort that could be put into the model development itself. The moderator could supply that knowledge, but it is well known that first hand knowledge is really required in order to address all the context-sensitive requirements of computational process models. An overlay on this debate was whether the developers would be required to model experienced operators or novice operators. There were strong arguments against modeling novices in very complex simulation environments, mostly centering on concerns that the likely variability they would produce in the data would mask the behaviors we were trying to measure. Using a task of realistic complexity also had implications for the Moderator team, which had limited resources for collecting data. Either they would have had to identify and recruit experienced operators from the domain under study, or invest in a very extensive period of training. Finally, high-fidelity, DoD-relevant simulation environments often are classified at a level that prohibits release in the public domain, and that would conflict with our goal of making all materials from the project available for use by others.

Having weighed these concerns, the Moderator opted to use a highly-abstracted version of an air traffic control (ATC) task and use participants who had played a lot of video games, but had no previous experience with this task. Stable data were obtained from novice human participants in four-hour sessions and the modelers were able to develop the requisite knowledge based on their own experience or by testing a small set of previously untrained participants themselves.

1.2.5 Experiment II: Category Learning

As the first comparison was wrapping up, the decision was made to focus the second comparison on learning. Considerable effort was devoted to meeting the same constraints considered for Experiment I. We wanted the task to be fairly abstract so that extensive content knowledge would not be required. It was important that participants have no previous exposure to the material to be learned and that they would not need extensive training to understand the task required of them. The resulting decision was to use the same basic air traffic control scenario already available from Experiment I, but to modify it to embed a category learning task. The learning task was based on the Shepard, Hovland & Jenkins (1961) classic category learning paradigm and its more recent replication and extension by Nosofsky, Gluck, Palmeri, McKinley, and Glauthier (1994). The availability of data from these classic experiments was considered a valuable feature of the task, because it allowed the modeling teams to get started on preliminary model developments using existing published results, while the Moderator made task modifications and collected the new human data. The second comparison challenged the modelers to build computational process models that simulated the learning of new concepts in the context of executing the air traffic control task, which is an interesting and novel dual task requirement. The second comparison also challenged them to make *a priori* predictions of human behavior in a transfer condition. The transfer data were actually withheld from the model developers until after they shared their predictions with the group. As the second comparison was wrapping up, the team decided to write a book about the AMBR project. We describe the book in the following section.

1.2.6 Preview of the Book

The book on which this report is based (Gluck and Pew, in press), is divided into three sections. Section I (Chapters 1-3) is background material leading up to the model descriptions. Chapter 1, obviously, is an overview of the effort. Chapter 2 describes, for each of the experiments, more about the rationale for the choice of tasks, a detailed description of the task, its dynamics, and the

human operator requirements. It then presents the method and results from the human experiments. Chapter 3 describes the hardware and software that were used and how the software was set up to allow seamless introduction of either a human operator or a model of the behavior of a human operator and the way in which the models were connected into the simulation.

Section II (Chapters 4-7) presents each of the models that were developed in response to the modeling challenges. The authors of these chapters were given a detailed structure to follow to assure that the chapters would cover similar topics and so that the reader would find it easier to follow the model descriptions and modeling results. At the end of each of these chapters the authors were asked to answer a set of summary questions about their models.

Section III is comprised of variations on conclusions, lessons learned, and implications for future research. Chapter 8 offers a discussion of how the models compared in terms of how the architectures and models were similar and different and how they performed the target tasks as compared with human data. Included are comments on how the results of the models' performances were related to and derived from the architectures and assumptions that went into the models. Chapter 9 relates the AMBR models of category learning to other models of category learning in the contemporary psychological literature. Chapter 10 covers a variety of important issues associated with the validation of computational process models. Chapter 11 is composed of reflections on the results of the project and proposes a research agenda to carry the field of human behavior representation forward. Chapters 1, 2, 3, 8 and 11 are included as part of this report.

A CD is included with the book. The primary content is loadable/runnable versions of the D-OMAR Air Traffic Control simulations and the human data that have been collected. Each modeler was asked to include, at a minimum, a readable text file for each model they developed, so interested persons can inspect the knowledge content and representation for each model. Additional functionality, such as a model that will actually load and run, is optional, at the discretion of the respective modeling teams. We hope this book and its supporting material will be an informative resource for all those interested in improving our human behavior representation capabilities.

1.2.7 References

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1993). *Rules of the mind.* Mahwah, NJ: Lawrence Erlbaum Associates.

Deutsch, S. E., & Benyo, B. (2001). The D-OMAR simulation environment for the AMBR experiments. In the *Proceedings of the 10th Annual Conference on Computer-Generated Forces and Behavior Representation*, 7-14. Orlando, FL: Division of Continuing Education, University of Central Florida.

Deutsch, S. E., MacMillan, J., & Cramer, N. L. (1993). *Operator Model Architecture (OMAR) demonstration final report (AL/HR-TR-1996-0161).* Wright-Patterson AFB, OH: Armstrong Laboratory, Logistics Research Division.

Gigley, H. M., & Chipman, S. F. (1999). Productive interdisciplinarity: The challenge that human learning poses to machine learning. In *Proceedings of the 21st Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates.

Gluck, K. A., & Pew, R. W., (Eds.) (in press). *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation*. Mahwah, NJ: Lawrence Erlbaum Associates.

Hart, S., & Staveland, L. (1988). Development of the NASA-TLX: Results of empirical and theoretical research. In P. Hancock and N. Meshkati (Eds.), *Human mental workload*. Amsterdam: North-Holland, 139-184.

Hollnagel, E. (1993). *Human reliability analysis: Context and control.* London: Academic Press.

Just, M. A., Carpenter, P. A., and Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping, 8*, 128–136.

Kanerva, P. (1993). Sparse Distributed Memory and related models. In M.H. Hassoun (Ed.), *Associative neural memories: Theory and implementation* (pp. 50-76). New York: Oxford University Press.

Kintsch, W., Healy, A. F., Hegarty, M., Pennington, B. F., Salthouse, T. (1999). Models of working memory: Eight questions and some general issues. In A. Miyake & P. Shah (Eds.) *Models of working memory: Mechanisms of active maintenance and executive control.* (pp. 412-441). New York: Cambridge University Press.

Morrison, J. E. (2004). *A review of computer-based human behavior representations and their relation to military simulations (IDA Paper P-3845).* Alexandria, VA: Institute for Defense Analyses.

Miyake, A., & Shah, P. (Eds.) (1999). *Models of working memory: Mechanisms of active maintenance and executive control.* New York: Cambridge University Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Nosofsky, R.M., Gluck, M., Palmeri, T.J., McKinley, S.C., & Glauthier, P. (1994a). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory & Cognition, 22*, 352-369.

Pew, R. W., & Mavor, A. S. (Eds.) (1998). *Modeling human and organizational behavior: Applications to military simulations*. Washington, DC: National Academy Press.

Pylyshyn, Z.W. (1991). The role of cognitive architecture in theories of cognition. In K. VanLehn (Ed.), *Architectures for Intelligence*. Hillsdale: Lawrence Erlbaum Associates Inc.

Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R. M., Gobet, F., & Baxter, G. D. (2003). *Techniques for modeling human performance in synthetic environments: A supplementary review (HSIAC-SOAR-2003-01)*. Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center.

Shepard, R. N., Hovland, C. L., & Jenkins H. M. (1961). Learning and memorization of classifications, *Psychological Monographs, 75* (13, Whole No.517).

Vidulich, M. A. & Tsang, P. S. (1986). Collecting NASA Workload Ratings: A Paper-and-pencil Package (Version 2.1), Working Paper. Moffett Field, CA: NASA Ames Research Center.

Young, R. M., Barnard, P., Simon, T., & Whittington, J. (1989). How would your favorite user model cope with these scenarios? *SIG CHI Bulletin, 20*(4), 51-55.

2. The AMBR Experiments: Methodology and Human Benchmark Results

Yvette J. Tenney, David E. Diller, Stephen Deutsch, Katherine Godfrey

This chapter describes two experiments in which performance data were collected from humans as a benchmark for comparing the ability of four different modeling teams to replicate and predict the observed data. Our goal was to stress and extend existing modeling architectures by collecting a rich set of data that would require models to successfully integrate and coordinate memory, learning, multi-tasking, cognitive, perceptual, and motor components. The first experiment focused on multiple task management and attention sharing. The second experiment expanded upon the first, embedding a category learning paradigm in a multi-tasking paradigm.

An unusual feature of the AMBR program was our development of an experimental testbed in which both humans and model participants could function. The testbed was instrumented so that a model could "perceive" the same events as a human participant, and had the ability to perform the same actions as its human counterparts. All actions were time stamped and recorded for later analysis. The details of this environment are described in Chapter 3 (this report).

In this chapter, we describe the experimental task and the human benchmark results, which served as the "ground truth" for assessing the performance of the models. For a detailed discussion of how well each of the models fared in predicting the observed human data see Gluck and Pew (in press). In chapter 4 (this report), we take a broad look across models, illustrating where they produce results similar to one another, as well as where they make their own unique predictions. It is these similarities and differences, as highlighted in model performance, that raise questions that can help us better understand the processes by which we as humans operate effectively in complex tasks.

2.1 Experiment 1: Integrative Multi-tasking

The initial experiment of the AMBR project was designed to examine human multiple task management, dynamic priority setting, and attention management as the modeling foci because these areas represented capabilities that were not widely available in existing models or modeling architectures and because they will be very important to future computer-generated forces (CGF) representation.

A simplified air traffic control situation was used because of the potential for human operator overload and the need for effective information management strategies. The goal was to foster

understanding of multi-tasking strategies, a capability not widely available in existing models, while providing a relatively straightforward task for initiating the model comparisons.

2.1.1 Overview

Decision making in complex, fast-paced environments has been studied through simulations in a number of domains, including air traffic control (e.g., Ackerman & Kanfer, 1994; John & Lallement, 1997;Macmillan, Deutsch, & Young, 1997), military command and control (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, in press; Cannon-Bowers & Salas, 1998) and team sports (Kirlik, Walker, Fisk, & Nagel, 1996). An Air Traffic Control (ATC) task developed by Macmillan, Deutsch, and Young (1997) was adapted for purposes of the present study. The focus was on individual rather than team performance. The task was loosely based on that of a real air traffic controller, but was designed to require minimal participant training. It was chosen because it exhibited the characteristics typical of mentally and perceptually demanding, multi-task situations: information arrives at inconvenient or unexpected times, information interrupts an ongoing chain of thought, information relevant to one task may be obscured by information from another, and information irrelevant to the current task may be salient and distracting. In addition, this task readily accommodated variations in display and workload conditions that were expected to have a predictable impact on performance.

The task, though simpler than its real-life equivalent, presented a host of challenges to the modelers. For example, they had to decide how the model would manage the scenario as a whole, how to choose when to shift between tasks, remember and update tasks waiting service, and prioritize among them. An additional challenge for the modelers was to ensure that the behavior of the model changed appropriately under different conditions, for example, with different display types and workload time pressures.

The task involved transferring aircraft in and out of a central sector by reading and sending messages to the aircraft and to the adjoining controllers. Penalty points were accrued for not carrying out actions within a critical time period, causing aircraft to go "on hold," and not attending to holding aircraft in a timely manner. Smaller penalties were accrued for skipping optional actions and for sending inappropriate, unnecessary, or inefficiently executed messages. The player's goal was to complete the scenario with a minimum of penalties. Optimal performance required staying ahead of the situation (e.g., anticipating the needs of aircraft

approaching a sector boundary), attending to high priority aircraft, remembering to complete actions following an interruption, and developing optimal scanning and reading strategies.

In the ATC task just described, the messages carry no ambiguity (i.e., requests could not be refused). As a result, the tasks across different aircraft and different flight stages all had a similar flavor. To help capture the greater variety that characterizes realistic multi-tasking situations, we introduced a decision task that took the form of a speed change request. Each time there was a speed request, the controller had to answer yes or no, depending upon the alignment of the aircraft. Penalty points were accrued for delayed or incorrect responses. The correct rule was explained at the outset in Experiment 1. In experiment 2, using a similar task, participants had to discover the correct response (i.e., learn the concept) on the basis of feedback.

Each participant in Experiment 1 experienced six scenarios, derived from a combination of three workload levels (2, 3, or 4 aircraft per minute) and two display conditions—a text condition in which all messages had to be read and a color condition in which color codes signaled the action required and obviated the need for reading. The purpose of the experiment was to see to what extent human performance differed as a function of workload and display conditions and to see how well the models would replicate the human results.

2.1.2 Method

2.1.2.1 Participants

Sixteen BBN Technologies employees, four females and twelve males, participated in the experiment. All were experienced video game players under the age of thirty-six.

2.1.2.2 Display

Participants were presented with a visual display consisting of a simulated radar screen, six action buttons, and three message boards with each board associated with two of the action buttons (see Figure 1). The simulated radar screen consisted of a central sector, bounded by yellow lines, representing a 200 x 200 nautical mile (NM) region. A "+" marked the center of the sector. The sector had both an outer and inner border (yellow and green lines), with 25 NM between the borders. The radar display includes four graphic icons representing neighboring controllers and may contain icons for the aircraft moving through the sector. Adjoining controllers (simulated) are located above, below, right, and left of the central sector, and are represented graphically by the words: *North*, *South*, *East*, and *West*.

The icons for the aircraft identify their direction of flight and are labeled with their flight designators (e.g., NW301). Aircraft traveled at a speed of 1,000 NM/hr or .28 NM/second, taking 3.6 seconds to travel one mile. Aircraft could enter and exit from all sides of the sector, flying across the sector on straight vertical or horizontal paths. Both incoming and outgoing messages appear over time on one of three message boards located beneath the appropriate actions buttons (see Figure 1). The aircraft labels within the text message boards are printed in a different color to make them stand out. Within each board, a message can be either left-adjusted or indented. The left-adjusted messages serve as triggers for actions. Indented messages reflect actions already taken and can be ignored, unless the player wants to check whether an action is still pending. New messages appear below previous text.

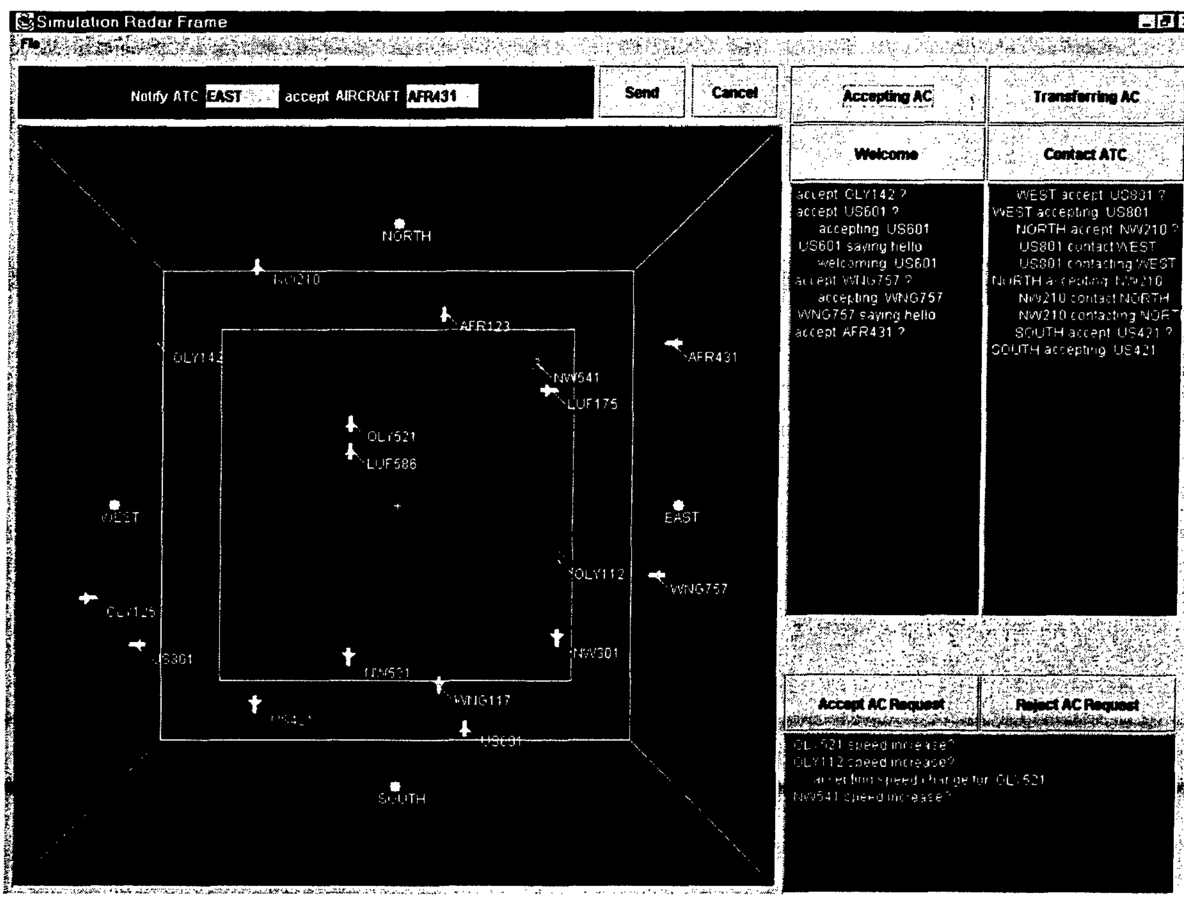


Figure 1. The ATC workplace.

Participants received one of two possible display conditions. In the text display the information required to determine what actions to take appeared in text form on the message boards on the

right side of the display. In the color display (see Figure 1), the aircraft icons themselves were color-coded to indicate required actions.

2.1.2.3 Design

The design was a within-participants comparison of two display conditions (text, color) and three workload levels (low, medium, high). Workload was manipulated by keeping the number of aircraft and their speed constant, but reducing the length of the scenario, by judiciously spacing aircraft closer together in the high workload scenarios. In this way, the number of possible penalty points remained constant across workload conditions. There were twenty aircraft requesting a hand-off and three requesting a speed change in every scenario. These requests arrived over a period of ten minutes for the low workload condition, seven and a half minutes for the medium workload, and five minutes for the high workload condition. The actual scenario length was 11.5, 9, and 6.5 minutes, for the three conditions, respectively to allow time for all requests to be processed.

Four equivalent sets of scenarios were developed for the experiment. Two sets of scenarios (*A* and *B*) were constructed, each with a random assignment of aircraft starting location, aircraft identification labels, and start times in order to make the sets as equivalent as possible. Two additional scenario sets (A* and B*) were generated by rotating each aircraft in the original scenarios by 180 degrees and randomly reassigning aircraft labels. For example, an aircraft that would enter from the northern portion of the west sector in the original scenario *A*, would enter from the southern portion of the east sector in the new scenario, A^*. Half the participants were trained on the *A* and A^* scenarios and tested on the *B* and B^* scenarios. The other half were trained on the *B* and B^* scenarios and tested on *A* and A^*. Within each group, half the participants received the unstarred scenarios in the text display condition and the starred scenarios in the color condition. The other half received the starred scenarios in the text condition and unstarred in the color condition. The starred and unstarred versions, mirror images of each other, were structurally equivalent and therefore, directly comparable in difficulty.

2.1.2.4 Task Activities

In the experimental task, the "player" (human participant or cognitive model) assumes the role of the Air Traffic Controller in the central sector bounded by the outer yellow line (see Figure 1). The player is responsible for managing the aircraft as they enter and leave the central sector. An aspect of this ATC task that sets it apart from the real one is that collisions are not of concern.

Players are told that aircraft that appear to be colliding are actually at different altitudes. The objective of the task is to complete the required actions in a timely fashion and to avoid accumulating penalties for missed, delayed or incorrect actions.

There are six actions that players can take by using the six action buttons (see Figure 1). An action is initiated by pressing one of the action buttons. For aircraft coming into the central sector, the player should ACCEPT and WELCOME each aircraft. For aircraft within the sector, the player should reply to a request for a speed increase by using the ACCEPT/REJECT AC REQUEST buttons. For aircraft leaving the sector, the player can TRANSFER the aircraft to the next controller and tell the aircraft to CONTACT ATC. Each of these actions and the penalties accrued by incorrectly performing or not performing these actions is described in further detail below and in Table 2.

Pressing an action button brings up a message template in the upper left hand corner, above the radar screen with slots indicating the required information (e.g., aircraft label and ATC). Template slots are filled in by selecting the appropriate icons on the radar display. The template can help the player avoid the penalty for extraneous clicks by serving as a reminder (e.g., only half the commands require clicking on ATC).

Table 2: Penalty Points in the Experiment 1 ATC Task

Penalty Category	***Player's Goal***	***Penalty***
Hold	Prevent aircraft from holding either while incoming or outgoing	50 points each time an aircraft turns red
Holding Delay	Get aircraft out of holding	10 points for each time unit[a] aircraft stays red
Speed Error	Respond to speed change requests correctly	50 points for an incorrect response to a speed change request
Speed Delay	Respond to speed change request in timely manner	2 points for each time unit[a] request not answered
Welcome Delay	Welcome aircraft in a timely manner	1 point for each time unit[a] aircraft not welcomed
Duplication	Avoid sending the same message twice.	10 points for duplication of a message
Extraneous Click	Avoid clicking on an air traffic control center when not required	10 points for an extraneous click
Incorrect Message	Avoid sending a message when proper trigger not present	10 points for an incorrect message

[a] *The time unit was 60 seconds for Low Workload, 45 seconds for Medium and 30 seconds for High Workload scenarios, respectively, to keep the maximum number of penalty points that could be earned in each condition constant.*

Accept. When the aircraft is 25 miles outside the outer boundary (yellow square), a message appears on the left-most board: (e.g., "ACCEPT TWA555?"). In the color display condition the aircraft icon also turns green. The player must ACCEPT the plane as soon as possible after the message appears.

If the player does not ACCEPT an aircraft before it reaches the outer border, the aircraft will turn red and enter a holding pattern. The player can release the aircraft by clicking on the aircraft and doing an ACCEPT. There are penalties associated with an AC turning red and staying red (see Table 2). Aircraft will turn red in both the text and color display conditions.

Welcome. Some time after an aircraft has been accepted, a message appears on the left-most board (e.g., "TWA555 saying hello."). In the color display condition the aircraft also turns blue. The WELCOME action is an optional action. Omitting it will not cause the aircraft to turn red. There are small penalties, however, associated with a delay in welcoming an aircraft.

Accept/Reject request. Aircraft within the bounds of the central sector may, from time to time, request a speed increase. A message appears in the bottom-most message area (e.g., "TWA555 speed increase?") and in the color display condition the aircraft turns magenta. Players can respond with an ACCEPT AC REQUEST or a REJECT AC REQUEST action. Participants are instructed that the judgment of whether to accept or reject the request for a speed increase is entirely straightforward and does not require any calculation of speed or distance. If the aircraft requesting a speed increase has no aircraft traveling in a direct line in front of it, the players must ACCEPT AC REQUEST, otherwise they must REJECT AC REQUEST. An incorrect response to a speed request carries a heavy penalty; a delay in responding to the request carries a lighter penalty.

Transfer. This action is the only one that is not triggered by a message, but rather, by the position of the aircraft. When an aircraft in the central sector reaches the inner border (green line), the player should initiate a TRANSFER to hand the aircraft off to the controller in the next sector. In the color display condition the aircraft icon turns brown.

Contact ATC. As soon as the next controller accepts the aircraft, a message appears on the right board: (e.g., "EAST accepting TWA555."). In the color display condition the aircraft icon turns yellow. If both actions, TRANSFER and telling the aircraft to CONTACT ATC, are not completed by the time the aircraft reaches the outer boundary, the aircraft will turn red and enter a holding pattern, with an ensuing penalty. The aircraft can be released by carrying out the

missing TRANSFER and/or CONTACT actions. The player may have to read the messages to determine if a TRANSFER, a CONTACT, or both, are required (assuming this information is not remembered). If the player responds to a red outgoing aircraft by doing a TRANSFER and it turns out the TRANSFER had already been carried out, the player accrues a small penalty for a duplicate action. There is a penalty for delaying the release.

In the color display condition the icon color turns back to white when the SEND is completed. If the SEND is pressed at the last minute, just before the aircraft reaches the border, the aircraft may turn red for a few seconds before turning white. The red indicates that a penalty has been accrued. (Note: In both the color and text displays, the assignment of a penalty in this situation depends on when the simulated adjoining controller actually receives the message and has time to act on it, which can vary depending on how busy the controller is.)

2.1.2.5 Procedure

Each participant took part in two sessions scheduled no more than a few days apart. The first session, 2-1/2 hours in length, involved an initial phase of training and a practice block of six blocks (covering 2 display conditions x 3 workload levels). The initial training began with the text display for all participants. The initial training phase included a demonstration, coached and uncoached practice with simple and complex scenarios, written figures and diagrams, a short quiz to ensure that material was understood, and, finally, the practice block, which served as a dress rehearsal for the actual test blocks.

The second session, 2 hours in length, involved the actual test block of six scenarios. The block consisted of three scenarios with one display, followed by three with the other. Half the participants started with text and half with color. Within each display condition, the workload level increased over the three scenarios. A single practice scenario, with the text display, preceded this block. Performance measures on the practice and test blocks were collected and compiled automatically during and after each run.

At the end of each scenario, for both the practice and test blocks, participants completed the unweighted Task Loading Index (TLX) workload rating sheet (Vidulich & Tsang, 1986). Ratings are made by circling a tick mark on a drawing of a 10-unit scale (yielding rankings from 0 to 10) on six different workload scales (mental demand, physical demand, temporal demand, performance, effort, frustration). TLX was selected because, of the two most widely used

subjective assessment tools, TLX (Hart & Staveland, 1988) and SWAT (Reid & Nygren, 1988), TLX is the easier to administer and was potentially easier to manage conceptually in a model.

A debrief questionnaire concerned with the participant's strategies was administered at the conclusion of the experiment (Pew, Tenney, Deutsch, Spector, & Benyo, 2000). Participants were asked about the difficulty of using the two displays, to what extent they read the text messages under different conditions (color/text display when busy/not busy), their strategies for scanning the radar screen, and how much they relied on their memory vs. other means for determining what actions to take when an outgoing aircraft turned red (i.e., TRANSFER, CONTACT ATC, or both).

2.1.2.6 Procedure for Human Performance Models

Data from the first eight human participants was given to the modelers during model development (development data), while data from the second half of the participants was being collected (intended comparison data). The original plan of using data from the unseen half of the subjects for the comparison had to be abandoned, however, because of the variability related to the small sample size. Instead, the models were compared against the full, more reliable data set, even though it was not entirely new to them. The procedure for the models was the same as for the humans, with one exception: Models did not answer debrief questions, although, as an extra challenge, they were encouraged to provide workload ratings.

2.1.3 Results

The human results, concerning performance, reaction time, and strategies will now be discussed. Chapter 4 (this report) provides a comparison of the model results and their architectures.

2.1.3.1 Accuracy Measures

Figure 2 illustrates the human data with respect to mean accumulated penalty points by condition. Error bars represent dual-sided 95% standard error of the mean confidence intervals. The graph clearly shows that more penalty points were accrued with the text display than with the color display, especially at higher workloads. These trends were supported by a two-factor repeated measures analysis of variance, with significant main effects of display [$F(1, 14)=23.27$, $p < .001$], and workload [$F(2, 28)=10.80, p < .001$] and a significant interaction of Display x Workload [$F(2, 28)=9.76, p < .001$].

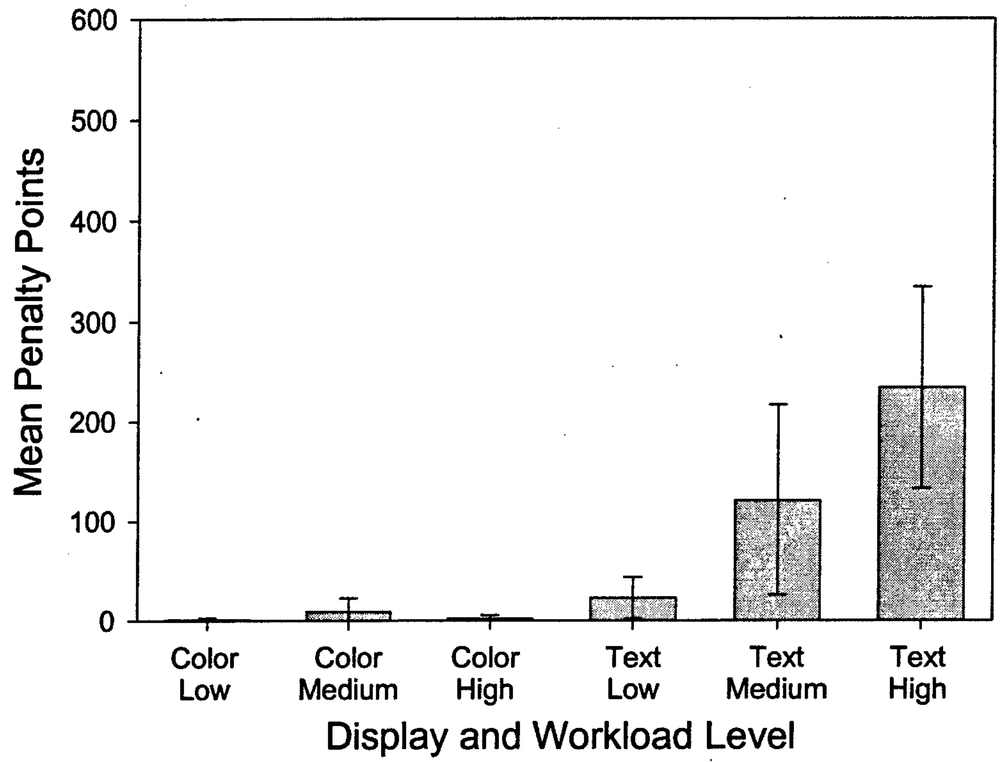


Figure 2. Penalty scores as a function of display and workload.

Penalty scores were explored in greater detail in the most demanding condition: text display with high workload. The upper panel in Figure 3 shows the penalty points in each of the penalty subcategories for the text-high workload condition. It is clear from the graph that the overriding source of points for humans was hold penalties (at 50 points each).

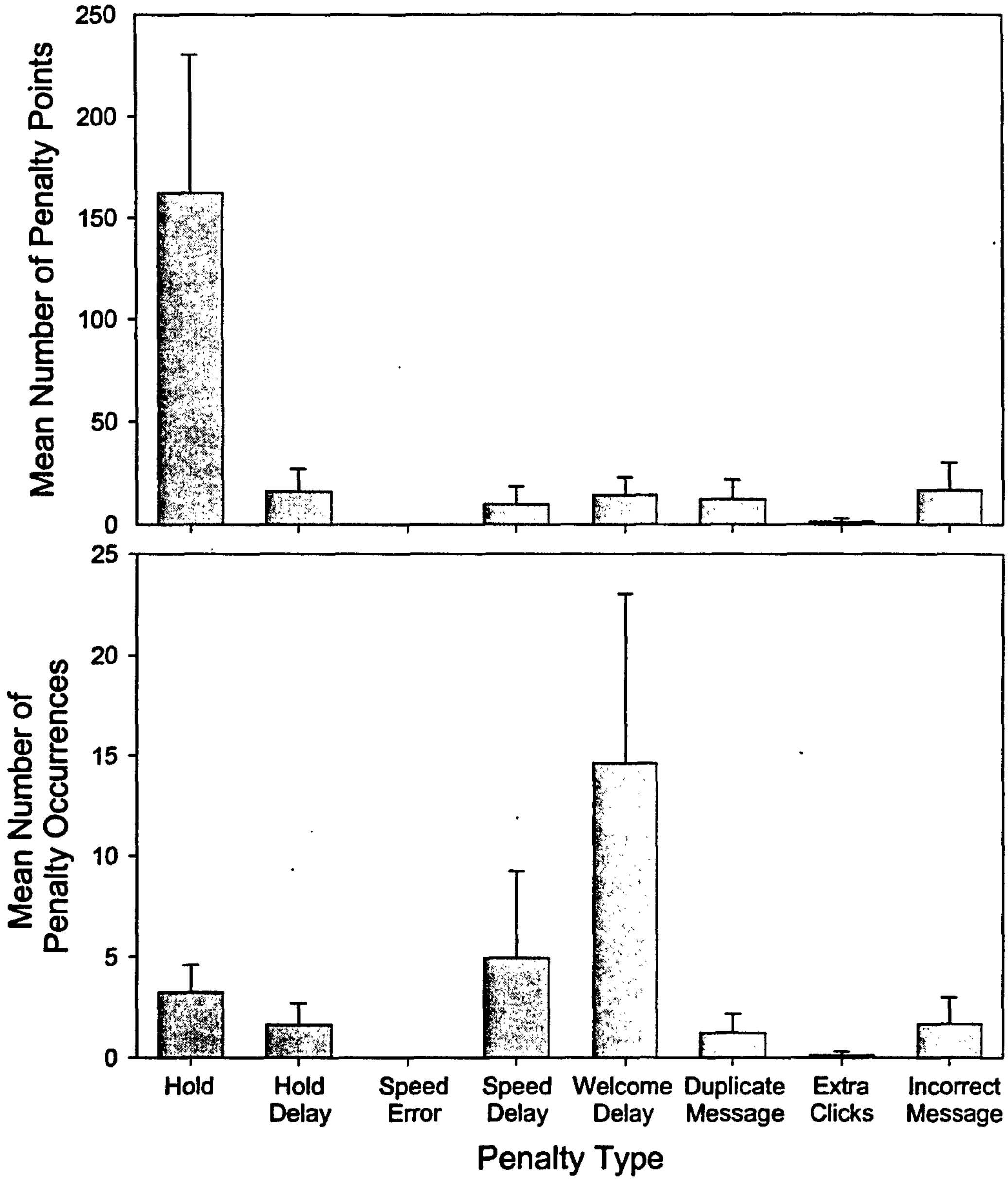


Figure 3. Detailed analysis of penalty categories for text-high workload condition.

The lower panel in Figure 3 shows the actual number of occurrences of each type of error. The results suggest that participants prioritized their actions so as to minimize overall penalties. Thus, welcome delay, which carries the lowest penalty (1 point per minute), was the most frequent penalty obtained by humans. The next largest category of observed errors was speed delay (2 points per unit of time). The strategy of postponing actions carrying low penalties to focus on preventing aircraft from turning red, which carries a higher penalty (50 points), is a reasonable strategy for coping with high workloads.

2.1.3.2 Response Time Measures

Figure 4 illustrates the mean response times for each display and workload condition. Response times were calculated as the time interval between the appearance of the trigger for an action and the activation of the SEND button to complete that action. All types of actions were included in the average.

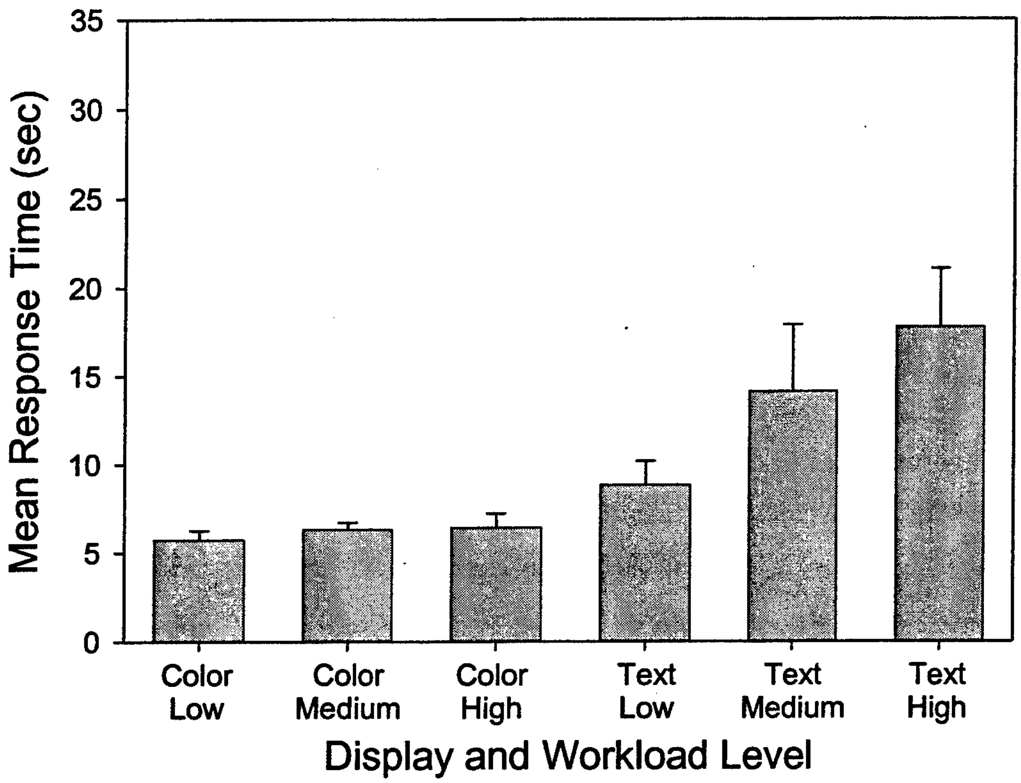


Figure 4. Mean response times as a function of display and workload.

As can be seen in the graph, participants responded to the triggers more quickly with the color display than with the text display, and workload effects were more pronounced in the text than in the color condition. These results show a similar pattern to the results seen in accuracy measures, suggesting there was no speed accuracy tradeoff occurring for the conditions. An analysis of variance showed significant main effects of display [$F(1, 14)=60.78, p <.0001$] and workload [$F(2, 28)=19.69, p <.0001$], as well as an interaction of Workload x Display [$F(2, 28)=12.73, p <.0001$].

2.1.3.3 Workload Measures

An overall subjective workload rating was obtained for each participant by averaging across the six individual workload scales that are part of the TLX (mental demand, physical demand, temporal demand, performance, effort, frustration). The workload rating scale ranged from 0 to 10 representing low to high workload, respectively.

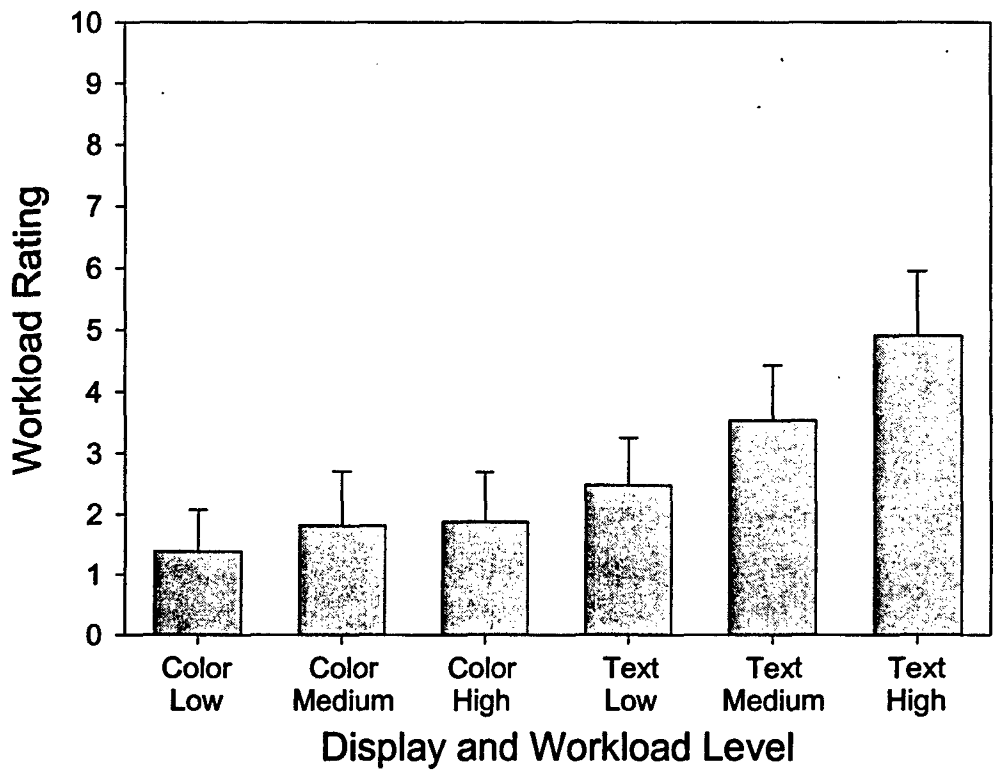


Figure 5. Subjective workload as a function of display and workload condition.

The human results, shown in Figure 5, demonstrate that participants rated their workload as higher for the text than for the color display. There was also an increase in subjective workload as actual workload increased, especially for the text display. An analysis of variance showed significant main effects of display [$F(1, 14)=43.97, p < .0001$] and workload [$F(2, 28)=24.43, p < .0001$] and a significant interaction of Display x Workload[$F(2, 28)=13.21, p <.0001$].

2.1.3.4 Debrief Questionnaire for Human Participants

The debrief questionnaire provided insight into human strategies and experiences. Many of the strategies reported by the human participants are reflected in the rationale presented by the modelers for their modeling decisions (see chapters 4-7 in Gluck and Pew, in press).

All participants rated the unaided text display as more difficult than the color-coded display (Question 1). The average rating on a scale of 1 to 10, where 1 was very easy and 10 was very difficult, was 7.5 (range 4-10) for text and 1.94 (range 1-4) for color.

Participants reported adjusting their strategy for accomplishing the task when they switched from one display to another (Question 2). The following answer was typical: "Yes, where color was involved [I] only had to watch for color changes and take appropriate action. No color - had to keep mental 'list' of items in queue." Another response alluded to a possible loss of situation awareness with the color display: "When it got busier, I relied more on colors than knowing which planes were at what point in their trip."

Participants were asked to rate how much they looked at the messages on a scale of 1 to 5 (never, rarely, sometimes, often, always) under various conditions (Question 3). Their responses indicated that they rarely looked at the messages in the color display, whether busy (1.8) or not busy (2.4), whereas they often looked at them in the text condition, both when busy, 4.06, and when not busy, 4.13.

Almost three-quarters of the participants answered "yes" when asked if they had scanned the radar screen in a consistent manner (Q4). Of those participants, almost half mentioned scanning in a "clockwise," or "north, east, south, west" pattern. Almost all participants mentioned focusing attention on the critical boundaries: the green and yellow borders, the area between the two borders, the area just inside the green border, the area just outside the yellow border.

A final question concerned strategies for when an aircraft turned red (Question 5). Participants were asked to indicate on a scale of 1 to 5 (never, rarely, sometimes, often, always) how often they engaged in a particular strategy. Their responses indicated that participants "sometimes" (3.1) remembered what they had already done for that plane and knew immediately what action(s) needed to be taken.

When they did not remember what they had already done for that plane, they "sometimes" (3.2) scanned the list of messages to see which action(s) they had omitted, "rarely" (2.4) ignored the message screen and instead did a CONTACT ATC to see if the red would disappear, and "rarely" or "sometimes" (2.5) ignored the message screen and immediately did a TRANSFER AC, followed, in due time, by a CONTACT ATC.

2.1.4 Discussion

This experiment was, in many ways, successful in producing a domain and dataset suitable for evaluating and comparing human performance models. The experimental paradigm provided a relatively rich dataset, although a larger sample of participants might have reduced some of the observed variability and inconsistencies seen in the data. Unfortunately, the experimental conditions were such that the color display condition produced very little challenge to the human participants, and resulted in a less challenging condition than had been desired. With respect to reaction time measures, reaction times increased with workload level. In addition, response times in the color display conditions were faster than the easiest text display condition. With respect to the main focus of this experiment, multiple task management, the results showed evidence of load shedding, with more occurrences of welcome delays and speed delays than of holds. Finally, subjective workload ratings clearly reflected workload condition. For a discussion of how successfully each of the different human performance models predicted all these results, see chapter 4 (this report) as well as chapters 4-7 in Gluck and Pew (in press).

2.2 Experiment 2: Category Learning

Another challenging avenue for model development, exploration, and comparison was sought for Experiment 2. We decided to focus on category learning in the context of a dynamic multi-tasking environment, given that two of the architectures (COGNET/iGEN and DCOG) did not yet have a learning component, while the other two, ACT-R and EASE (through its predecessor SOAR), had already been applied to category learning phenomena. We were curious to what extent each of the models would borrow from or reuse existing approaches to categorization and adapt them for multi-tasking purposes. The large literature and rich set of findings associated with concept learning made this capability appealing. The classic category learning study by Shepard, Hovland and Jenkins (1961) and its more recent replication and extension by Nosofsky, Gluck, Palmeri, McKinley, and Glauthier (1994a) have served as benchmarks against which most models with category learning capabilities are compared (Anderson, 1991; Gureckis & Love, 2003; Kruschke, 1992; Nosofsky, Palmeri, & McKinley, 1994b). While there are a large number of category learning models, the majority of them addresses category learning in isolation, rather than integrated with other behavioral phenomena. By including a variant of the classic study by Shepard, et al. (1961) in the air traffic control task from Experiment 1, we hoped to develop a complex task with a rich set of findings against which to compare the models.

2.2.1 Overview

In the study by Shepard, et al. (1961), participants were asked to classify eight stimulus items, varying on three binary valued dimensions (size, color, shape). Stimulus items were organized into two categories, with four items in each category. Given these constraints, there are six possible category structures, illustrated in Figure 6, with the eight stimulus items represented by the eight numbered circles at the corners of the cube. Category assignment is represented by the filled and unfilled circles. Every possible assignment of stimulus items to categories falls into one of the shown category structures, or problem types. The main finding of Shepard, et al. (1961) was that the six problem types varied in their difficulty to learn, with Problem Type I, the easiest to learn, and Problem Type VI, the most difficult to master.

Problem Type I requires information about only one dimension. Problem Type II requires knowledge of two dimensions, and is the exclusive-or (X-OR) problem with an irrelevant dimension. Problem Types III, IV, and V require information from all three dimensions, with varying degrees of relevance. Problem Type VI requires information from all three dimensions and places equal importance on all dimensions. The results from Shepard, et al.'s (1961) original study and Nosofsky, et al.'s (1994a) replication and extension found that Problem Type I was learned most easily, followed by Problem Type II, then by Problem Type III, IV, and V, which were approximately equal in difficulty, and lastly Problem Type VI. Nosofsky, et al. (1994a) tested a larger number of participants than in the original Shepard et al. study, collecting enough data to produce learning curves, and provide insights into the time course of category learning.

We embedded this classic category learning task in the basic air traffic control situation, couching it as an altitude change request from an aircraft pilot. Participants had to learn to make correct decisions to accept or reject altitude change requests, based on three bivariate properties of the aircraft (percent fuel remaining, aircraft size, and turbulence level). In addition to the altitude change requests (the concept learning task), the participant had to hand off a number of aircraft to adjoining controllers (secondary task), similar to what they had done in Experiment 1.

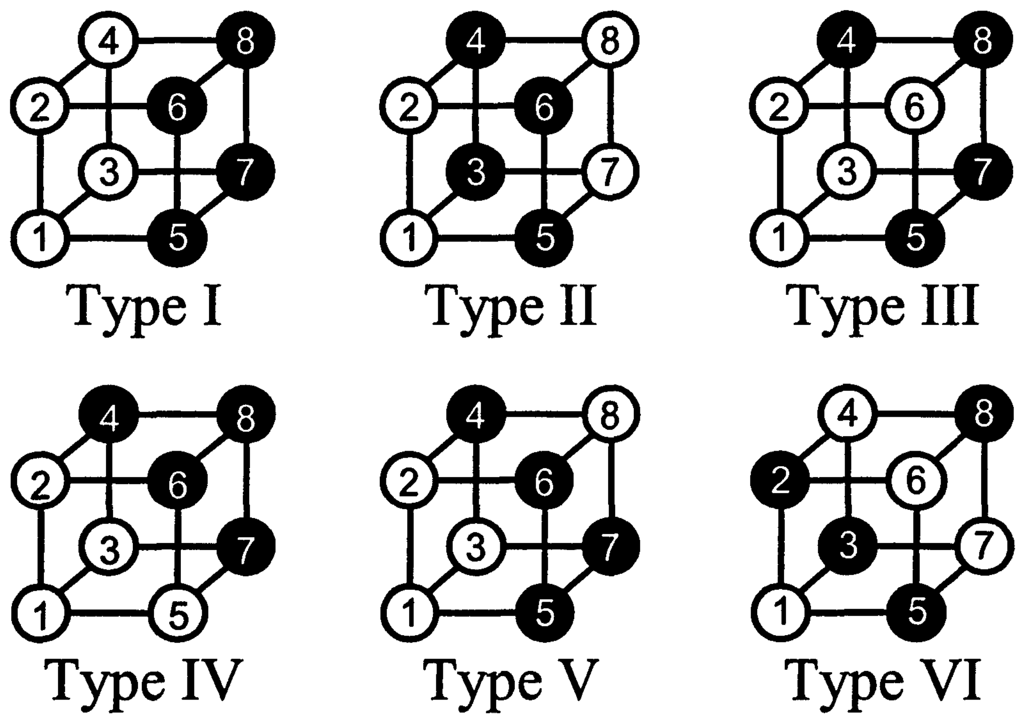


Figure 6. Logical structure of the six types of problems tested by Shepard, et al. (1961).

Concept learning also lent itself well to the addition of a transfer test, something our Expert Panel had been recommending as critical for model validation. Another advantage was that psychologists studying categorization had developed debriefing techniques for inducing subjects to verbalize their strategies. We felt this kind of information would be valuable to the modelers (Love & Markman, 2003).

The learning requirement provided an opportunity to compare the different ways models could be made to expand their capabilities. By adding a category learning component to the air traffic control task, modelers were required to either activate and adapt learning algorithms already existing in their models, or in several cases, develop an entirely new learning mechanism. The degree to which learning mechanisms were integrated into existing model architectures could vary from a separate sub-module implemented as a 'black box' and able to be manipulated independently from the rest of the system, to a fundamental component and constraint on the model architecture itself. We were interested in the different approaches the models would take and how successful they would be in matching human learning.

2.2.2 Method

2.2.2.1 Participants

Participants were ninety college undergraduates (mean age 21.75, range 18-33) with at least sophomore status and an intended or declared psychology major. Seventy-three percent were female. Fifty-four students were recruited from colleges in the Boston area, predominantly from Boston University, and were tested at BBN Technologies in Cambridge, MA. Thirty-six additional participants were recruited from and tested at the University of Central Florida in Orlando. Participants were randomly assigned to conditions. One of the BBN experimenters observed several testing sessions in Florida to ensure that identical procedures were followed. One participant was replaced due to an inability to understand the secondary airspace management task, indicated by a lack of responsiveness to a large number of task action cues. As a non-native English speaker, it is possible she had not understood the instructions because of insufficient mastery of English.

2.2.2.2 Display

The visual display and stimulus items were adapted from the color display condition in Experiment 1 (The text display was not used in this experiment). Two changes were made to the color display. First, three properties of the aircraft were on occasion displayed on the simulated radar screen just below the aircraft label. These three properties represented percent fuel remaining, aircraft size, and turbulence level, each with one of five possible values. Possible values for percent fuel remaining were "10", "20, "30", "40", or "50". Values for aircraft size were "XS", "S", "M", "L", or "XL," while values for turbulence level were "0", "1", "2", "3", or "4". Of the five values for each property, only two, the second and the fourth, appeared in the learning portion of the experiment. The remaining three values were reserved for the transfer task, described below. Second, the action buttons previously labeled ACCEPT AC REQUEST and REJECT AC REQUEST were relabeled ACCEPT ALTITUDE REQUEST and REJECT ALTITUDE REQUEST, respectively.

2.2.2.3 Design

The design consisted of two between-participants factors each containing three levels and one within-subjects factor. The two between factors were category problem type and workload level; the within factor was blocks. Three different problem types (i.e., category structures) were used, and were identical to problem types I, III, and VI used in Shepard, et al. (1961) and in Nosofsky,

et al. (1994a). The logical structures of the three problem types are shown in Figure 6. Three workload levels, low, medium, and high, were explored, each with a different number of required secondary task actions. Participants were randomly assigned to one of these nine groups.

2.2.2.4 Procedure

Participants completed eight presentation blocks or scenarios. This procedure was similar to that used in Nosofsky, et al. (1994a). Each of the eight scenarios contained sixteen category judgment requests, with each stimulus item appearing twice per block. The stimulus ordering within each block was randomized. The workload level and problem type remained constant over the eight blocks.

The task and procedures were based on the task and procedures used in Experiment 1, with several extensions. A category learning component was added to the task, as was a transfer condition, completed at the end of the eight blocks. The category learning task was emphasized as the primary task and is described below. The multi-tasking airspace management portion of Experiment 1 was treated as a secondary task, and is also discussed below.

Primary Task. The category learning task was couched as a request from a pilot to change altitude. Altitude change requests were used in place of speed change requests from Experiment 1. Just like speed requests, altitude requests were signaled by an aircraft turning magenta and by a message appearing in the bottom-most message area (e.g., "UAL250 altitude change?"). Participants were instructed that their main task was to determine the correct responses to altitude change requests based on three properties of the aircraft. These properties were displayed at the same time the aircraft turned magenta, and remained until either feedback was provided for a response to an altitude change request or until a 30 second deadline for responding had been exceeded. If a participant did not respond to the altitude change request within fifteen seconds, a warning was provided of the impending time limit. This warning consisted of the aircraft icon flashing and all secondary task action buttons grayed out and unusable until the participant responded. Participants not responding prior to the deadline incurred a penalty of 200 points, while making an incorrect response only accrued 100 points, so it was advantageous to make a guess rather than not respond. Participants were warned that they might also incur additional penalties by not being able to perform needed actions during the time the secondary task buttons were grayed out.

Feedback was given after each response in both visual and auditory form. Feedback to an incorrect response consisted of a low "buzz" tone presented for 350 msec as well as an "X" icon presented next to the aircraft label. Positive feedback consisted of a high chime sound presented for one second and a 'smiley face' icon. The visual feedback icons were presented for five seconds.

Secondary Task. The secondary task involved handling aircraft that are either entering or leaving the central sector in the same manner as in Experiment 1. This time however the length of the scenario was held constant and the number of hand-off requests was varied to create different workload conditions. Each of the eight blocks/scenarios contained sixteen aircraft with the number of these aircraft requiring airspace management actions by the participant varying by workload condition. In the high workload condition, all sixteen aircraft made a hand-off request with eight entering and eight exiting the airspace. In the medium workload condition, twelve hand-offs were required with half entering and half exiting the airspace. In the low workload condition, no aircraft required handoffs, but all made altitude change requests. To summarize, high workload scenarios consisted of thirty-two requests (sixteen altitude changes and sixteen hand-offs), medium workload consisted of twenty-eight requests (sixteen altitude changes and twelve handoffs), and low workload scenarios consisted of sixteen requests (sixteen altitude changes and no handoffs). Each of the scenarios lasted ten minutes.

Penalty point structure. The penalty point structure was identical to that used in experiment 1, with the exception that speed errors and speed delays were replaced with penalties for incorrectly responding or not responding to the altitude change request. An incorrect response to the primary task (altitude change request) garnered 100 penalty points, while failure to respond to the altitude change request within the allotted time earned 200 penalty points. Failure to respond to the secondary task (hand-off request) carried lower penalties. This penalty point structure ensured that participants gave priority to the primary, category learning task and did not skip any category judgments. This procedure did, in fact, result in a complete set of learning data from each participant, even in the most difficult condition.

Transfer Task. Following the training phase, a transfer task was conducted to provide insight into what participants had learned and retained from the training phase (see Table 3). The transfer test consisted of 25 items, 8 old and 17 new. The 8 old items (trained) were identical to those in the training phase. Eight of the new items (extrapolated) had values for all three properties that

were more extreme than the values presented during the training phase and therefore could be responded to by analogy with the trained stimuli. The remaining nine new items (internal) had a value for one or more of the properties that was halfway in between the values during the training phase. They were included to force participants to make new classification decisions, but are not included in these analyses, since there were no clear predictions for these items.

Table 3: Aircraft Properties during Training and Transfer Phases

	Stimulus Properties			
Phase of Experiment	% Fuel Remaining	Size	Turbulence Level	Example Items (3 properties)
Training	20, 40	S, L	1, 3	20 L 1
Transfer	10, 20, 30, 40, 50	XS, S, M, L, XL	0, 1, 2, 3, 4	50 L 2

Table 4 shows the complete set of eight different training items followed by the twenty-five transfer test items. The left column (Items) shows the value for the three properties of the aircraft (percent fuel remaining, aircraft size, and turbulence level). The numbers 1-5 in this column refer to the five possible values for each property, as shown in Table 3. The letters A and R, in the top half of Table 4 refer to the binary responses 'Accept' and 'Reject' that are required by the structure of the different problem types. The particular items that were to be accepted vs. rejected, in the training phase, were counterbalanced across subjects.

Table 4: Structure of the Training and Transfer Task Items

Training Phase

	Problem Type		
Items	*I*	*III*	*VI*
2 2 2	A	A	A
2 2 4	A	A	R
2 4 2	A	A	R
2 4 4	A	R	A
4 2 2	R	R	R
4 2 4	R	A	A
4 4 2	R	R	A
4 4 4	R	R	R

Transfer Phase

Item Set	*Item*
Trained	222, 224, 242, 244, 422, 424, 442, 444
Extrapolated	111, 115, 151, 155, 511, 515, 551, 555
Internal	133, 233, 433, 533, 313, 323, 343, 353, 333

The aircraft were all simultaneously present in the display and did not move. A static display was used to allow ample time for participants to consider each new stimulus item. Participants were told that they would be seeing some new property values in addition to those they had seen before and were instructed to make their decisions to accept or reject an altitude change based on what they had learned previously. In a randomly chosen sequence, each aircraft, one at a time, turned magenta, and the three aircraft property values were presented on the screen. Participants judged whether to accept or reject each aircraft based on the property values. The presentation was self-paced, with the aircraft reverting to white and its properties disappearing after the aircraft was judged. No feedback was given during the transfer task.

Procedure for Human Participants. Participants were tested individually by one of two experimenters in Cambridge or one of three experimenters in Florida. The experiment took about four hours: an hour and a half for instruction and practice and two and a half hours to complete the tasks and debriefing. At the onset of the experiment, participants were given a color vision test to ensure they could differentiate the colors used in the experiment. Participants were trained

in the general task through the viewing of several short instructional videos illustrating the main parts of the display and the possible actions and penalties. After viewing the videos, participants responded to several short training scenarios designed to help them understand the mechanical characteristics of the task, after which participants filled out an on-line, multiple-choice quiz to ensure that they had understood the instructions. Correct answers were displayed to any incorrect responses and participants were retested until they achieved a perfect score on the test.

After completing the quiz, participants took part in the eight learning scenarios. Each scenario lasted ten minutes. A workload questionnaire was completed online at the beginning, middle, and end of the training, after Blocks 1, 4, and 8. The questionnaire was similar to the one used in Experiment 1, but used a seven-point rating scale, and the following format: Question 1: Mental Demand: How mentally demanding was the task? Question 2: Physical Demand: How physically demanding was the task? Question 3: Temporal Demand: How hurried or rushed was the pace of the task? Question 4: Performance Errors: How likely were you to make mistakes on this task? Question 5: Effort: How hard did you have to work to accomplish your level of performance? Question 6: Frustration: How insecure, discouraged, irritated, and annoyed were you?

Following the eight blocks, participants were given the transfer task, which was self-paced. Finally, human participants completed an online questionnaire designed to elicit details about their learning strategies on the primary task (see Table 5).

Table 5: Debriefing Questionnaire

Question	***Response Type***
Screen 1	
On the last of the eight blocks (the ones with the moving planes and smiley faces):	
1. How did you decide whether to accept or reject an altitude change request?	Open Ended
2. Did your strategy change over the 8 blocks [Please explain]	Open Ended
Screen 2	
On the last of the eight blocks (the ones with the moving planes and smiley faces):	
1. Did you use a rule? (check) Yes No	Yes/No
2. If yes, I accepted an altitude request when:	Open Ended
3. If no, what did you do?	Open Ended

2.2.2.5 Procedure for Human Performance Models

Early in the model development cycle, we shared the data from the human learning phase, including primary and secondary task performance, reaction times, workload ratings and debrief responses, with the modeling teams. These data were provided to facilitate the modeling efforts. However, in order to test the model's ability to predict, and not simply replicate, human behavior, the results of the transfer task were not revealed to the modeling teams until after an initial round of model predictions was produced by the modeling teams. Modeling teams were then provided with the results of the transfer task and allowed to revise their models in light of these results. Results for the human performance models are presented individually in chapters 4-7 (see Gluck and Pew, in press), and compared as a whole in chapter 4 (this report).

2.2.3 Results and Discussion

We had originally planned to analyze the results in terms of three variables: problem type and secondary task workload (between-participants) and blocks (within-participants). However, preliminary analysis revealed, surprisingly, almost no effect of the workload manipulation. While there were significantly more hand-off errors in the high than in the medium workload condition, confirming that the manipulation had in fact made the hand-off task more difficult, there were no significant effects of the secondary task on the category learning task, for either accuracy or response time. For this reason, the workload variable has been omitted from the analyses reported here. We speculate on possible reasons for the lack of a workload effect below.

2.2.3.1 Primary Task

Accuracy Measures. The category learning data are shown in Figure 7. The graph shows the mean probability of error for each block of 16 categorization judgments in each of the Type I, III, and VI problems. It is clear that in spite of the difference in the domains, the human results closely replicate those of previous studies (Nosofsky, et al., 1994a; Shepard, et al., 1961). The fewest number of errors occurred for the Type I problem, followed by the Type III problem. The Type VI problem showed the greatest occurrence of errors.

A two-factor mixed analysis of variance, with problem type as the between-participant variable and blocks as the within-participant variable yielded a significant main effect of problem type [$F(2, 87) = 23.47, p < .0001$] and of blocks [$F(7, 609) = 56.70, p < .0001$] and no significant interaction of Problem Type x Blocks[$F(14, 609) = 1.31$, ($p > .10$]. A Tukey pairwise contrasts

test ($p < .05$) confirmed that the three problem types were each acquired at significantly different rates from each other.

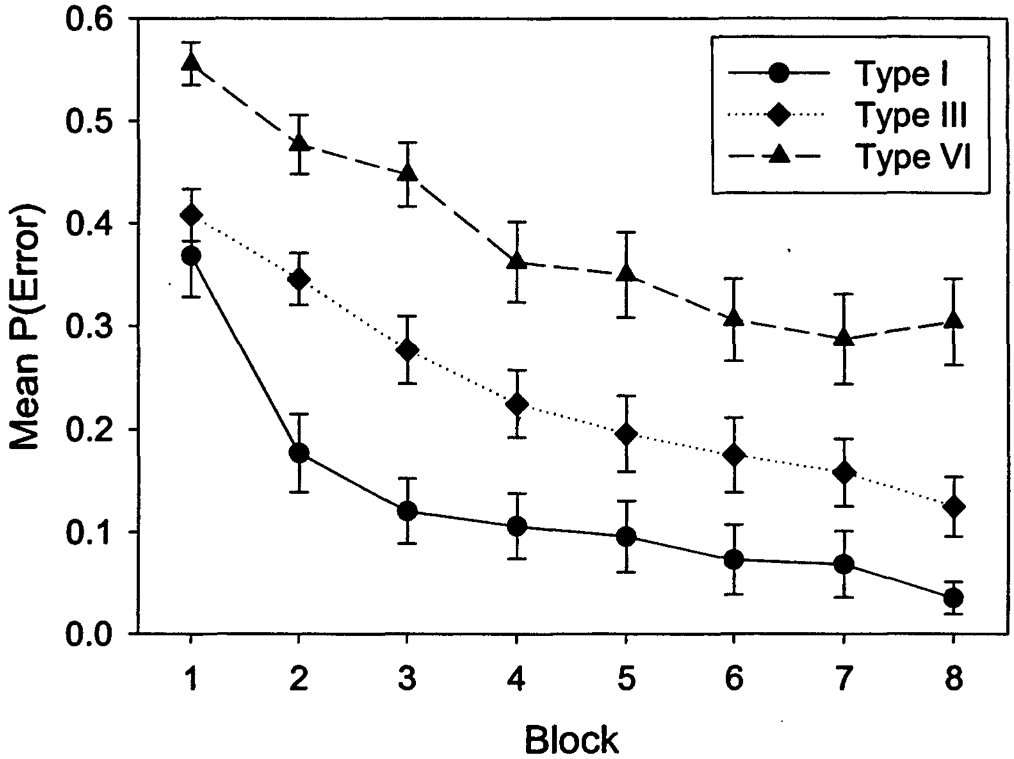


Figure 7: Category learning data for Type I, III, and VI problems.

A more fine-grained analysis of the data for Problem Type III is shown in Figure 8. In Problem Type III, there are two sets of stimulus items, where each item in the set has the same structural relationship to other items and is logically equivalent. Four stimulus items, 1, 2, 7 and 8 (see Figure 6, Problem Type III), are members of what can be described as the "central" set, since they share at least two features values with other members of the category, while items 3, 4, 5, and 6 are members of the "peripheral" set. Members of these sets are logically equivalent, meaning they can be interchanged with one another by rearranging the stimulus dimension labels. Because of their logical equivalence, it is possible to aggregate the responses to these stimulus items. Another way to think about the distinction between central versus peripheral sets is in terms of rules. Problem Type III can be thought of as a rule with two exceptions. If the rule is "accept if dimension 1 is maximal," items 4 and 6 are exceptions. If the rule is "accept if dimension 2 is maximal, items 3 and 5 are exceptions. Items 1, 2, 7, and 8 are members of the "central" set since they are never the exceptions, while items 3, 4, 5, and 6 are members of the "peripheral" set because they can be exceptions.

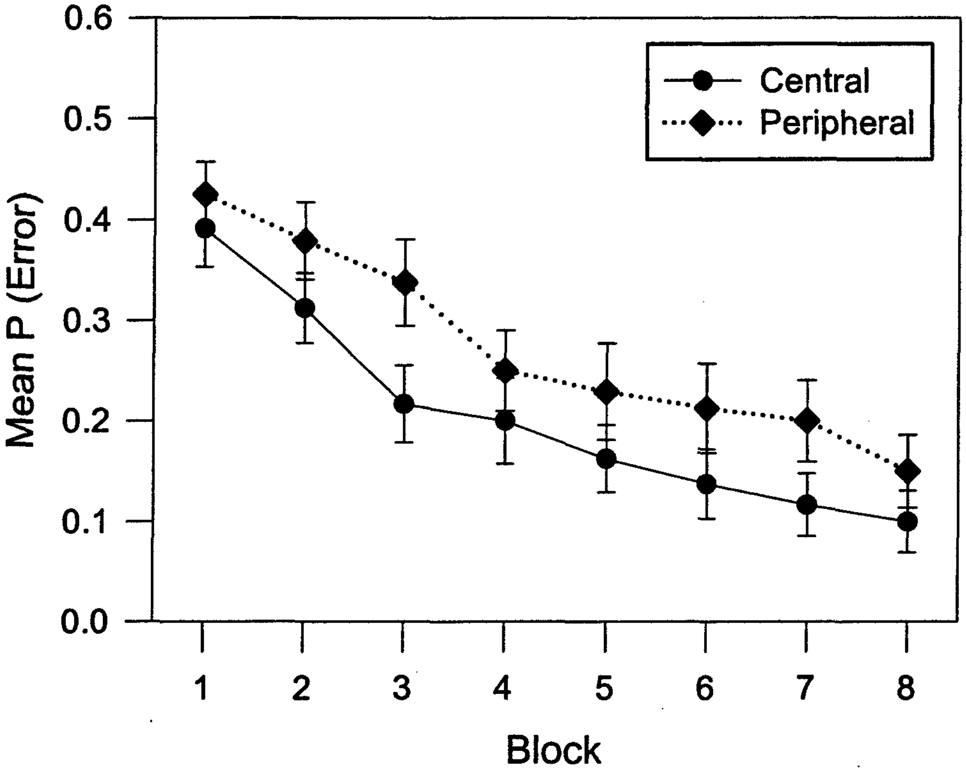


Figure 8: *Category learning data for the Type III problem learning data.*

Replicating the results of Nosofsky, et al. (1994a) for Problem Type III, the data showed that members of the central set were learned more quickly than peripheral items. A two-factor repeated measures ANOVA with item (central vs. peripheral) and block as within-subject variables, resulted in significant main effects of item [$F(1, 29) = 6.00, p < .05$)] and block [$F(7, 203)$, $= 20.50, p < .0001$] with no interaction of Item x Block [$F(7, 203) < 1.0, p > .10$].

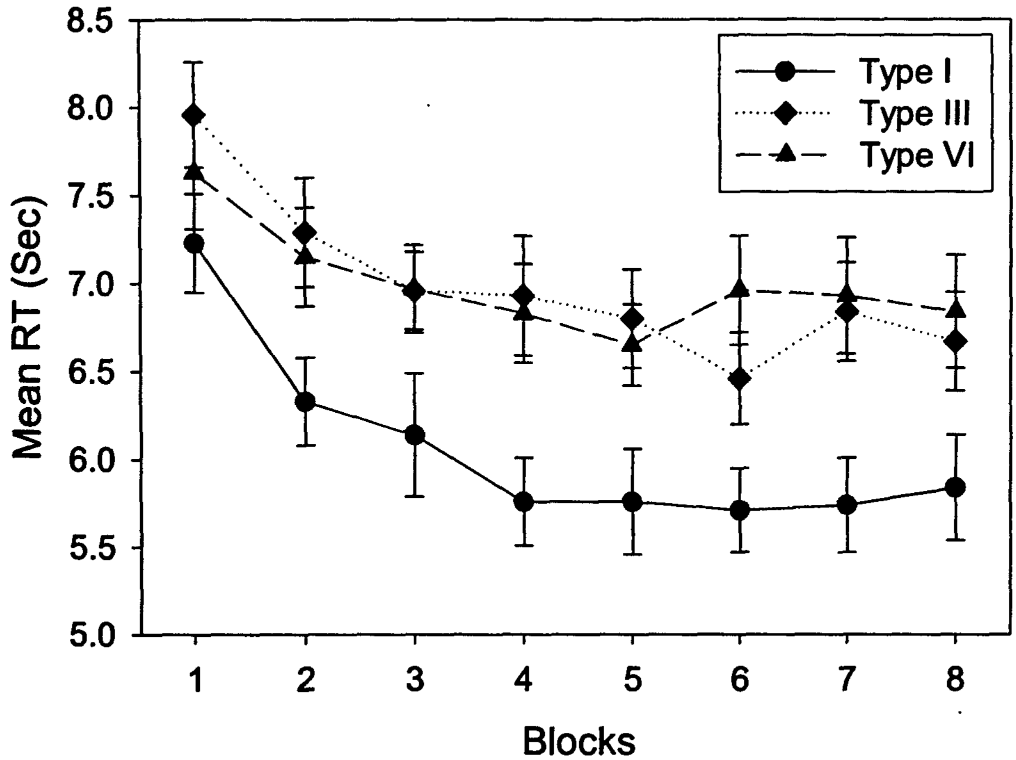


Figure 9. Response times to the category learning task as a function of category learning problem type.

Response time measures. Figure 9 shows the mean response times to the primary category learning task. The pattern of results was similar to the accuracy results. Again, there were significant effects of problem type [$F(2, 87) = 4.52, p < .05$] and of block [$F(7, 609) = 24.14, p < .0001)$], with no significant interaction of Problem Type x Block [$F(14, 609) = 1.28, p > .10)$]. A Tukey test revealed that responses were faster in Problem Type I than Problem Type III or VI ($p < .05$), which did not differ from each other.

2.2.3.2 Secondary Task

Penalty score measures. An analysis of variance of the penalties on the secondary task showed no significant effect of problem type [$F(2, 57) < 1, p >.10$], block [$F(7, 399) < 1, p >.10$] or Block x Problem Type [$F(14, 399) = 1.21, p > .10$]. The mean penalty score was low, 10.84, suggesting that performance on the hand-off task was quite accurate. Surprisingly, accuracy was not affected by the difficulty of the problem type on the category task, as would be expected if participants were multitasking and shedding the hand-off task when the primary, category learning task became more difficult. Evidently neither task was sacrificed for the sake of the

other. Difficulties caused by harder problem types in the altitude request task or more planes to transfer in the hand-off task did not spill over into the other task, as had been expected.

Response time measures. Response times for the secondary task again showed no main effect of the primary task. [$F(2, 57) < 1, p > .10$]. There was a significant main effect of blocks [$F(7, 399) = 2.83, p < .01$], with participants responding more quickly on later blocks (See Figure 10). However, this decrease in response time was not affected by the difficulty of the category task, as would be expected if participants were sacrificing time on one task to work on the other. The non-significant interaction of Problem Type x Block [$F(14, 399) = 1.68, p > .10$] suggests that participants were simply becoming more familiar with the hand-off task, rather than benefiting from the improved performance across blocks on the category task. The average response time for a hand-off was fairly fast (8.5 seconds) considering that three to four clicks of the mouse were required.

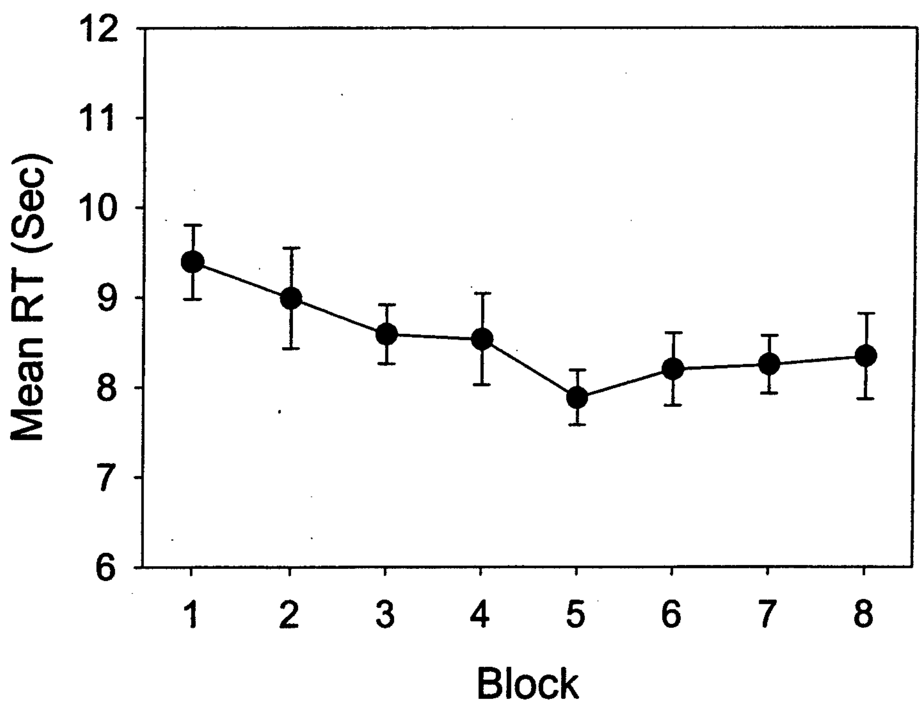


Figure 10. Response times to the secondary task as a function of blocks.

There are several possible reasons for the lack of multi-tasking within our paradigm. One is that the secondary workload levels may not have been sufficiently high to cause interference on the category task. We may have simply failed to find the "sweet spot" that would lead to task interference on the category task. Alternatively, our procedure may have discouraged multi-

tasking, and not facilitated the use of spare time while performing the secondary task to work on the categorization task. In pilot testing, using a slightly different procedure in which aircraft properties were visible at all times rather than being extinguished after each response, participants did seem to be dividing their time between the two tasks, e.g., by looking at the aircraft properties and reviewing what had happened during a break in the secondary task. In the final version of the procedure, in which aircraft properties vanished as soon as a response was made, perhaps there was less incentive to time-share. This change in procedure (removing the properties shortly after the participant made a response) was incorporated to replicate more closely traditional concept learning paradigms.

2.2.3.3 Subjective Workload Ratings

Participants were asked to rate the required workload of the task in its entirety (i.e., both primary and secondary task components together) after Blocks 1, 4, and 8. The results, illustrated in Figure 11, were similar to those found for the error measures. An ANOVA showed a significant main effect of problem type [$F(2, 87) = 3.25, p < .0$], with higher workload ratings for Problem Types III and VI than for Problem Type I ($p < .05$). There was a main effect of block [$F(2, 174) = 39.32, p < .0001$], with workload ratings declining across blocks, and no interaction of problem Type x Block [$F(4, 174) = 1.08, p > .10$].

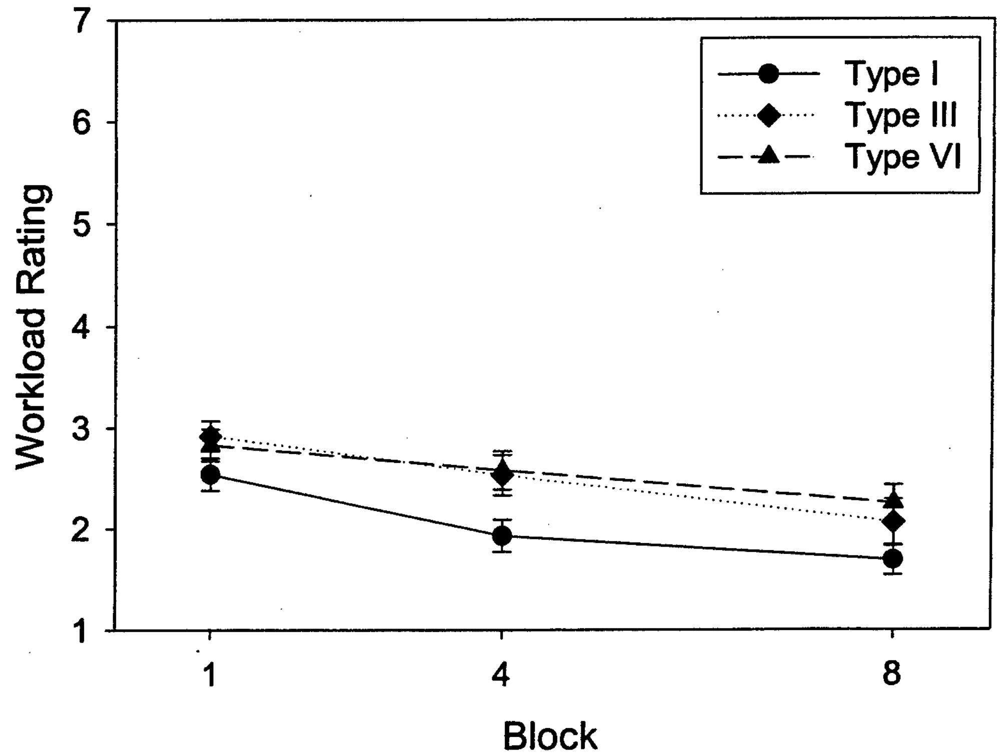


Figure 11. Subjective workload ratings administered after Blocks 1, 4, and 8.

2.2.3.4 Debrief Questionnaire

Responses to the debrief questionnaire were analyzed to determine strategies participants used for each of the three Problem Types on the primary task. We first analyzed whether participants reported using a rule during the last of the eight blocks, when asked a yes/no question pertaining to rule use (see Table 5, Screen 2). Eighty-one percent of the participants reported using a rule. Results mirrored the ease with which participants learned the different problem types. Participants in the Problem Type I condition reported using a rule most often (29 of 30) followed by Problem Type III (26 of 30) and lastly for Problem Type VI (18 of 30).

Secondly, we examined responses to the open-ended questions to determine which strategies were reported by participants during the last of the eight blocks (see Table 5, Screen 1). Most responses were indicative of rule use (77%), followed by a memorization strategy (18%) where participants reported remembering or memorizing the instances, and with a small number of participants reporting having guessed (3%). Two percent of the participants could not be identified as having any discernable strategy. Reports of memorization or guessing increased with problem type complexity. No participants indicated memorization or guessing strategies for

Problem Type I. Of the participants in the Problem Type III condition, six indicated memorizing the instances, one participant reported guessing, and the remainder, twenty-three, reporting rule-based strategies. The number of individuals reporting memorizing the instances increased to ten for Problem Type VI, with two participants indicating they were guessing, and the rest (eighteen) reporting rule-based strategies.

In order to better analyze the open-ended questions, we organized participants into *perfect* and *imperfect* learners. Perfect learners had achieved a perfect score on the category learning task on the last block. Perfect learner status was achieved by 80% of the participants in the Problem Type I, 50% in Problem Type III, and 30% in Problem Type VI. Of the perfect learners, 100% reported using a rule in Problem Types I and III, while 89% reported using a rule in Problem Type VI.

The striking finding about perfect learners was that they tended to report a limited set of common strategies, as described below. Non-learners rarely mentioned these strategies, and almost no one who mentioned one of them failed to achieve a perfect score. In other words, subjects' verbal reports of strategy use strongly correlated with performance at the end of the learning phase of the experiment.

For Problem Type I, all of the perfect learners reported using a 1-feature rule (e.g., "Accept if turbulence is 3"). Two of the imperfect learners also reported using a 1-feature rule, and may have simply discovered the rule at the last minute. The other imperfect learners tended to report rules that were more complicated than necessary, containing extraneous or incorrect features.

In Problem Type III, eleven of the fifteen perfect learners could be classified as using one of four strategies. Three of these strategies involved feature-based rules and one consisted of enumerating the four, presumably memorized, instances. The first strategy is a single feature rule with two exceptions, or instances. For example, accept small aircraft, except small planes with 20% fuel in turbulence level 3; also, accept large planes with 40% fuel remaining in turbulence level 3. The second strategy involves two 2-feature rules. For example, accept if aircraft is small in turbulence level 1 or the plane has 40% fuel remaining in turbulence level 3. This strategy is a clever alternate way of describing the structure inherent in Category 3. The third strategy is a 2-feature rule with two memorized instances. For example, accept small planes with 40% fuel remaining. In addition accept, "20 S 1" and "40 L 3". The last strategy reported was to memorize all four instances. Participants were judged as using a memorization strategy if they recited all

four exemplars in response to the question of how they decided which aircraft to accept. Interestingly, none of the imperfect learners in Problem Type III reported using any of these "winning" strategies. They tended to report partial, but insufficient rules, incorrect rules, or they said something vague, like "memorization," without going into detail.

In Problem Type VI, five of the perfect learners reported memorizing the instances. Four of these five reported four of the exemplars. The other four perfect learners reported using a correlated values rule. We define a correlated values rule as a rule in which participants recall two values as positively correlated, either two low or two high values, or negatively correlated, one high value and one low value. For example, accept large planes with high fuel and high turbulence or low fuel and low turbulence; also accept small planes with low fuel and high turbulence or high fuel and low turbulence.

2.2.3.5 Transfer Task

We begin our analysis of the transfer data by first comparing performance on the transfer items that had previously been encountered (transfer trained items) with performance on those same items from the last block of training (training Block 8 items, see Figure 12). We contrast these results with performance on extrapolated transfer items more extreme than the trained items (transfer extrapolated items). Extrapolated items were scored in the same manner as the nearest previously trained item. The transfer trained vs. transfer extrapolated comparison was designed to assess how well strategies generalized from one type of item to another. The training Block 8 vs. transfer trained comparison allowed for an evaluation of how well performance transferred from the learning portion of the experiment to the transfer condition.

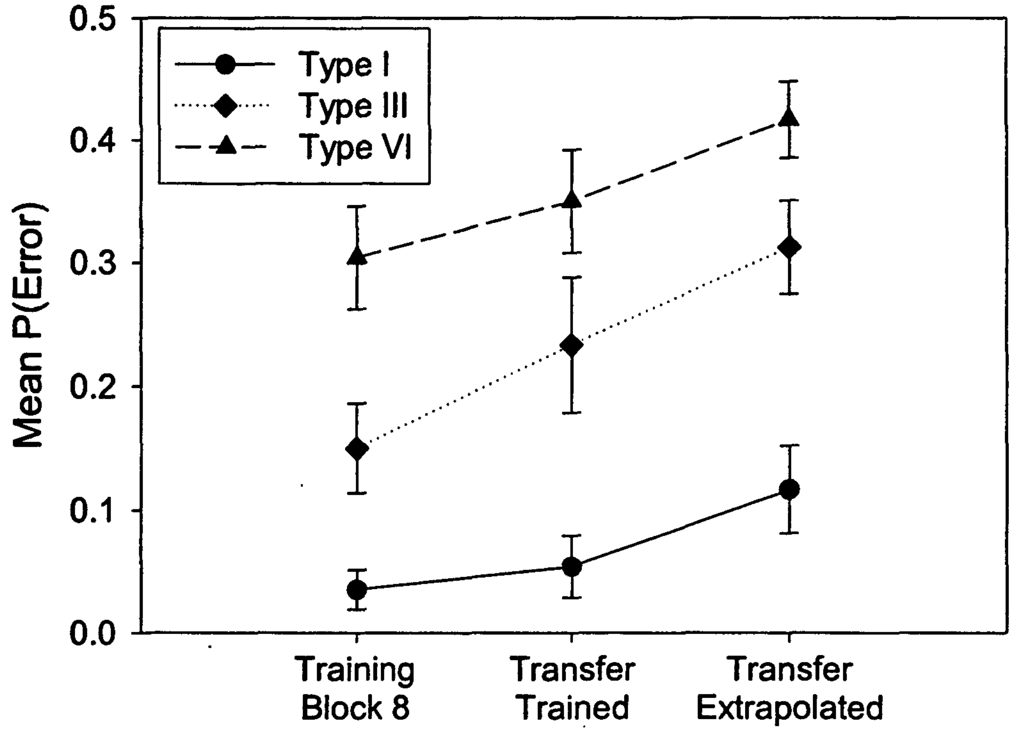


Figure 12. Transfer task results for Block 8 learning data, and trained and extrapolated transfer test items.

A two-factor mixed ANOVA, with problem type as the within-participants factor and items (training Block 8, transfer trained, transfer extrapolated) as the between-participants factor, showed there were significant main effects of problem type [(F2, 87) = 26.96, $p < .0001$] and items [($F(2, 174) = 18.91, p < .0001$] and no interaction of Problem Type x Items [$F(4, 174) = 1.16, p > .10$]. The results showed a significantly greater number of errors on the trained items on the transfer test than on the identical items in training Block 8. Less surprising was the finding that extrapolated items were missed more frequently than trained items on the transfer test. A Tukey analysis showed that all three types of items differed significantly from each other ($p < .05$).

2.2.4 Conclusion

We included a classic study of category learning as part of a multi-tasking air traffic control task. The category learning component of Experiment 2 provided a replication of results found in earlier studies by Shepard, et al. (1961) and Nosofsky, et al. (1994a), in particular the finding that certain problem types could be learned more easily than others. The transfer task extended these

results by examining the generalization of category learning to a new context. The results showed that both context and item changes produced a deterioration in performance. Accuracy was forfeited when learners had to switch contexts, from training to transfer test, even for those items that were identical in the two contexts. Performance declined even more when the items were analogous, but not identical to those previously learned. The loss of information across contexts may have been accelerated by the need to generate additional rules (e.g., such as rounding up) for the internal items. These new rules may have interfered with the retention of the original rules.

A striking finding was pervasiveness of rules reported in the debrief session and the extent to which particular strategies correlated with success on the task. The rules reported by the successful learners were uniformly simple for Problem Type I, and became more complex and varied for Type III. For Type VI, some participants discovered a surprisingly complex rule involving positively and negatively correlated values. Another successful strategy involved realizing there were only four items to accept and committing them to memory. This strategy was not as obvious as it seemed, because there were sixteen items in each block (the eight items were each repeated). Many of the strategies reported by those who were unable to master the task in the time allotted resembled "buggy" versions of the winning strategies, e.g., incorrect rules and inaccurate memorization of instances.

Although we desired to produce an interaction between category learning performance and a secondary task workload manipulation, no evidence was found for timesharing between the primary and secondary task. Perhaps a more difficult workload manipulation or a more integrated task structure would have led to timesharing. Even without this finding, a rich set of learning and performance data were collected on which to extend, evaluate and compare the four models. These models' ability to replicate and predict the human results described in this section and the implications for an emerging and exciting discipline provide the theme for the remainder of this report. (See also Gluck and Pew, in press.)

2.3 References

Ackerman, P.L. & Kanfer, R. (1994). Air Traffic Controller Task CD-ROM database, data collection program and playback program. Office of Naval Research, Cognitive Science Program.

Anderson, J.R. (1991). The adaptive nature of human categorization. *Psychological Review, 98*, 409-429.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y (In press). An integrated theory of mind. *Psychological Review.*

Cannon-Bowers, J.A. & Salas, E. (Eds.). (1998). *Making decisions under stress: Implications for individual and team training*. Washington, DC: American Psychological Association.

Gluck, K. A.., & Pew, R. W., (Eds.) (in press*). Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation* . Mahwah, NJ: Lawrence Erlbaum Associates.

Gureckis, T.M and Love, B.C. (2003). Towards a Unified Account of Supervised and Unsupervised Learning. *Journal of Experimental and Theoretical Artificial Intelligence*, 15, 1-24.

Hart, S., & Staveland, L. (1988). Development of the NASA-TLX: Results of empirical and theoretical research. In P. Hancock and N. Meshkati (Eds.), *Human mental workload.* Amsterdam: North-Holland, 139-184.

John, B.E. & Lallement, Y. (1997). Strategy use while learning to perform the Kanfer-Ackerman Air Traffic Controller task. *In Proceedings of the Ninteenth Annual Conference of the Cognitive Science Society.*

Kirlik, A., Walker, N., Fisk, A.D., & Nagel, K. (1996). Supporting perception in the service of dynamic decision making. *Human Factors, 38*, 288-299..

Kruschke, J.K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22-44.

Love, B. C., & Markman, A. B. (2003). The non-independence of stimulus properties in human category learning. *Memory & Cognition, 31*, 790-799.

MacMillan, J., Deutsch, S.E., & Young, M.J. (1997). A comparison of alternatives for automated decision support in a multi-task environment. *Proceedings of the 41st Annual Meeting of the Human Factors and Ergonomics Society.*

Nosofsky, R.M., Gluck, M., Palmeri, T.J., McKinley, S.C., & Glauthier, P. (1994a). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory & Cognition, 22*, 352-369..

Nosofsky, R.M., Palmeri, T.J., & McKinley, S.C. (1994b). Rule-plus-exception model of classification learning. *Psychological Review, 101*, 53-79.

Pew, R.W. Tenney, Y.J., Deutsch, S., Spector, S., Benyo, B. (2000). *Agent-Based Modeling and Behavior Representation (AMBR) Evaluation of Human Performance Models: Round 1 - Overview, Task Simulation, Human Data, and Results.* Cambridge: Distributed Systems & Logistics, BBN Technologies, A Verizon Company.

Reid, G. B., & Nygren, T. E. (1988). The subjective workload assessment technique: A scaling procedure for measuring mental workload. In P. A. Hancock & N. Meshkati (Eds.) *Human Mental workload.* New York: North Holland, 185-218.

Shepard, R.N., Hovland, C.L., & Jenkins H.M. (1961). Learning and Memorization of Classifications, *Psychological Monographs, 75* (13, Whole No.517).

Vidulich, M.A. & Tsang, P.S. (1986). *Collecting NASA Workload Ratings: A Paper-and-pencil Package (Version 2.1), Working Paper.* Moffet Field, CA: NASA Ames Research Center.

2.4 Authors' Note

We gratefully acknowledge the sponsorship of this research by the Human Effectiveness Directorate of the Air Force Research Laboratory. We thank AFRL Program Manager Mike Young and COTR Kevin Gluck for their guidance and support. We are extremely grateful to the modeling teams for their contributions to the AMBR project, including Bob Eggleston and Katherine McCreight (AFRL), Wayne Zachary, Jim Stokes and Joan Ryder (Chi Systems, Inc.), Christian Lebiere (CMU), and Ron Chong (George Mason University) and Robert Wray (Soar Technology, Inc.) We are indebted to Randy Astwood, and David Holness (Naval Air Warfare Center, Training Systems Division) and Sandra Spector (BBN Technologies) for their help in collecting human participant data.

3. The Simulation Environment for the AMBR Experiments

(Stephen Deutsch, David Diller, Brett Benyo, Laura Feinerman)

3.1 Introduction

One of the goals of the AMBR Model Comparison was to make the simulation environment for the comparison available to other researchers, so that they might have the opportunity to use it to extend our initial accomplishments in novel ways. This book and its accompanying CD accomplish that goal. Nevertheless, the development and use of computational process models and the simulation environments with which they will interact are challenging endeavors. Each simulation engine (such as that used in AMBR) has particular requirements and limitations, and these should be understood as well as possible before starting a new modeling and simulation effort. This chapter provides detailed information regarding the simulation environment used in the AMBR Model Comparison, and will serve as a useful resource for anyone considering the use of this software in future research or for those considering the development of new software for similar human behavior representation (HBR) research purposes.

3.2 D-OMAR Simulation for the AMBR Experiment

The Distributed Operator Model Architecture[3] (D-OMAR) served as the distributed simulation environment for the AMBR experiments. It provided the simulation environment for both the real-time human participant trials and the fast-time human performance model runs (Deutsch & Benyo, 2001). In the Experiment 1 Phase 1 trials and the Experiment 2 trials, socket-based native-mode connectivity linked the D-OMAR simulator to the simulators for the human performance models. In the Experiment 1 Phase 2 trials, native-mode connectivity was replaced by the HLA RTI for half of the trials.

3.2.1 The Scenarios for the AMBR Experiment Trials

The experiment trials were based on a simplified air traffic control environment where the human participant or human performance model played the role of an air traffic controller who was responsible for managing the aircraft in a sector and the transfer of aircraft to and from adjacent sectors. In Experiment 1, the modeling teams were challenged to build human performance models that reflected the management of multiple tasks and attention sharing as evidenced by the

[3] Detailed information on D-OMAR is available at http://omar.bbn.com.

human participants. In Experiment 2, the scenarios were revised to challenge the modeling teams with a concept learning task.

The starting points for the development of the AMBR experiment software were the scenarios (Deutsch & Cramer, 1998) that supported the MacMillan experiments (MacMillan, Deutsch, & Young, 1997). The MacMillan experiments had been implemented in an older all-Lisp version of the OMAR simulation system, hence software changes were required to update the scenarios to operate in the current D-OMAR simulation environment. The scenarios were then revised to meet the requirements of the new experiment designs.

Scenario scripts were developed for each of the AMBR experiment trials. Additional scripts were developed to support the training scenarios for the experiment participants. A detailed description of the content of these scenarios is contained in Chapter 2. The scripts defined the behaviors of the aircraft necessary to create the situations dictated by the experiment designs. They defined the timing and flight path for each aircraft and requests of the air traffic controller made by each aircraft.

In Experiment 1, scenario scripts were manually generated for the different experimental conditions. Aircraft position, routing, and velocity were chosen to produce the desired activities within the scenario, while other aircraft characteristics such as starting locations, aircraft identification labels, and aircraft start times were randomly assigned. For Experiment 2, a script generation program was developed. The script generator randomly constructed a set of scenario scripts for each participant in the experiment. Constraints were built into the script generator to ensure the scripts met the requirements for the each of the various experimental conditions.

3.2.1.1 The ATC Workplace

The ATC workplace for the AMBR experiment trials is shown in Figure 13. It includes a synthetic radar display and a message system to support communication among the players in the airspace—the air traffic controllers and the flight crews of the aircraft in the sectors.

The radar portion of the display includes icons for the aircraft and the neighboring controllers. The aircraft icons are labeled with flight designators (*e.g.*, NW301) and identify the aircraft's direction of flight. The icons for the neighboring controllers are labeled with their names (*e.g.*, EAST) reflecting their location with respect to the central sector. The aircraft and ATC icons are mouse sensitive and are used to provide slot values when generating messages. The sector boundaries for the air traffic control regions are painted in yellow (the outer square

and radial lines in Figure 13). The notification boundary, painted in green (the inner square in Figure 13), just inside the boundary for the center sector, marks the beginning of the region in which to initiate the transfer of aircraft to an adjacent sector.

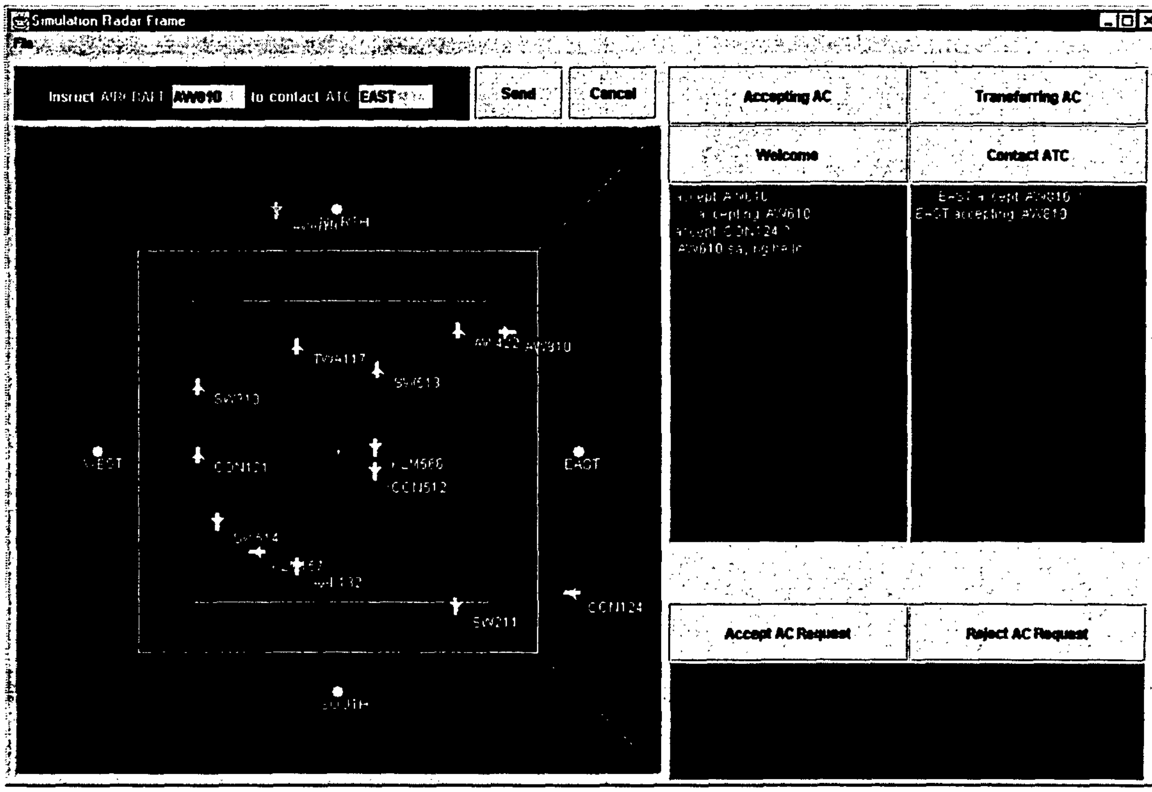


Figure 13: The ATC workplace.

The message system supports communication among air traffic controllers and between air traffic controllers and the aircraft in their sectors. A prompt line above the radar screen displays the current state of the message being prepared (see Figure 13). Screen buttons are available for message selection by type (*e.g.*, the button labeled "CONTACT ATC" in the upper right hand panel of Figure 13 initiated the message shown in the prompt line). Message slots are filled by using the mouse to select an aircraft or ATC icon from the radar screen. In the prompt line, the icons for the EAST controller and the aircraft AW810 have been selected. Using the mouse to select the "Send" button initiates the transmission of the message to the designated recipient. The message then appears in the appropriate message panel. A message can be canceled explicitly using the "Cancel" button or implicitly by selecting another message type.

For the AMBR experiments, messages were sorted to three separate screen panels by category (see Figure 13): those related to inbound aircraft appeared in the left-hand message panel), those related to outbound aircraft appeared in the right-hand message panel, and those

related to aircraft requesting speed changes (Experiment 1) or altitude changes (Experiment 2) appeared in the lower message panel. Messages requiring a response from the local controller (*e.g.,* an incoming request to accept an aircraft) were left justified within the panel, while those that did not require a response (*e.g.*, an outgoing message for an adjacent ATC to accept an aircraft) were indented.

In Experiment 1, there were two treatment conditions, the "text" condition and the "color" condition. Experiment 2 used only the color condition. In the text trials, participants had to watch for aircraft approaching the notification and sector boundaries and monitor the message screens to determine when actions were required. In the color condition, when an action was required, that aircraft's icon was painted in a color selected to indicate the specific action required. When the correct action was taken, the aircraft icon became white again. As expected, the text case proved significantly more challenging than the color case.

The Experiment 2 concept learning task was based on learning the correct response to an aircraft's altitude change request. As the request was made, the aircraft's color was changed from white to magenta. The experiment defined three situational properties, fuel remaining, aircraft size, and turbulence as the basis for the controller's decision to accept or reject the request. Single character alphanumeric symbols were presented beneath the aircraft's flight designator to denote the aircraft's current state for each property. A correct response was followed by a smiley face and a bell-like tone; an incorrect response was followed by an X and a growl-like tone. In the absence of a response within 15 seconds, the aircraft icon began to flash and non-altitude related responses were inhibited. At 30 seconds, the opportunity to respond to the request was terminated.

3.2.1.2 Model's View of the Workplace

An application-programming interface (API) was developed that defined the interface between the ATC workplace and the human performance models. The models' simulators were provided with information on each item as it appeared or moved on the ATC workplace screen. The form of "viewing" of the screen was determined by the model. The human performance models performed actions analogous to those performed by the human participants—viewing the radar and message panels of the ATC workplace screen, and constructing and sending of messages.

Basic static information such as map scaling, and the location of sector and notification boundaries, was made available to the model developers off-line. On-line updates took several

forms. The appearance of new aircraft was noted and aircraft positions were subsequently updated once per second. During the color trials, the models were notified of aircraft color changes as they occurred. The models were also notified of incoming messages and their content, and of the panel on which the message appeared. As out-going messages were constructed, information on the updating of the prompt line above the radar display was conveyed. For Experiment 2, aircraft parameter values were made available as altitude change requests were made and the models were notified of correct or incorrect responses and the erasure of the aircraft's parameter values shortly after their response.

Inputs at the workplace were generated by experiment participants: human participants or human performance models. They consisted of the mouse selects used to construct messages to adjacent controllers and sector aircraft. Message type was determined by a button select (*e.g.*, "ACCEPTING AC"). Slot values to complete the message were filled in by icon selects to identify an ATC (*e.g.*, "EAST" in Figure 13) and an aircraft (*e.g.*, "AW810"). Selection of the "Send" button initiated the transfer of the message to its designated recipient. As the message was sent, it appeared in the appropriate message panel.

3.2.1.3 The AMBR Scenario Agents

The principal agent in the AMBR scenarios, the experiment subject, played the role of the air traffic controller for the center sector. The controller was either a human participant or one of the four human performance models. When a human participant was acting as the center sector controller, a "lightweight" agent was used to track and record the actions of the human player. When a human performance model played the role of the controller for the center sector, the model typically, but not always, operated in its own simulator, in some cases operating on the same machine as the D-OMAR simulator and in other cases operating on a separate machine.

The remaining air traffic controllers, those for the four adjacent sectors, were played by D-OMAR human performance models. Aircraft transiting the sector were also active scenario agents, as were the radar systems provided for each of the sector controllers. The radar systems provided the information to drive the ATC workplace display for the experiment participant and the "displays" used by the D-OMAR models acting as controllers for the adjacent sectors. Air traffic controller models for the adjacent sectors used their radar systems to track aircraft and made use of the messaging system to handle communication with adjacent controllers and the

aircraft in their sectors. Aircraft used the messaging system in communicating with the air traffic controllers.

In real world air traffic control situations, exchanges between controllers and the aircraft in their sector take place at a rapid pace. Responses from the aircraft and adjacent sector models in the AMBR scenarios were consistent but leisurely. This was done to extend transaction times in order to create situations in which there were concurrent pending transactions—multitasking situations.

3.2.1.4 Automating the Experiment 2 Trials

Because of the fairly large number of participants involved in Experiment 2, human data were collected at two different facilities. In order to standardize the experimental process and minimize any procedural differences that might occur between the two different facilities, we automated much of the experimental process for Experiment 2. Most materials were presented and all information collected using the automated system, with the exception of the background form and some illustrations. Training materials, including movies with instructional voice-over, example scenarios, and a quiz were all part of the automated experimental process. Additionally, several questionnaires and the experiment debrief were performed online. These materials are on the CD that accompanies this book.

The experimenter began the experiment by launching the application and inputting the participant number for the session. The system then displayed the appropriate experimental session for the chosen participant. While some parts of the session were paced by the instructor or participant, others such as the videos and scenarios could not be interrupted or paused once initiated. All data collected during the experiment were automatically logged for offline evaluation. Although the system required both Lisp and Java processes to be initiated to run a scenario, the system was structured so that only a single application required launching by the experimenter.

3.2.2 D-OMAR Basics

D-OMAR was developed initially as a discrete event simulator with the flexibility to explore a range of architectures for human performance modeling. It has been used as a general-purpose simulation environment and more recently, it has been used extensively as the foundation for agent-based system development. Today there are two implementations of D-OMAR: the hybrid Lisp and Java implementation, now known as OmarL, and the newer all Java implementation,

OmarJ. The AMBR experiments were supported using OmarL; OmarJ was not available at the beginning of the AMBR project.

The basic elements of the D-OMAR simulation environment for the AMBR experiment trials are the:

- Simulation engine
- Scenario scripting capability
- Simulation control panel
- Application user interface
- Interface to the human performance models
- Data recording subsystem

Architecturally, the simulation engine, Core-OMAR is configured as a peer in a distributed computing environment. A D-OMAR simulation environment can be configured with a single Core-OMAR node for the entire environment, with multiple Core-OMAR peers in a distributed simulation environment, or for operation with one or more heterogeneous simulators.

The two principal elements of Core-OMAR, implemented in Lisp, are the simulator and the representation languages used in developing the scenarios that the simulator executes. The Core-OMAR simulator is an event-based simulator that can be run in either real-time or fast-time modes. The representation languages include a frame language, a rule language, and a procedural language. The Simple Frame Language (SFL) is a classical frame language derived from KL-ONE (Brachman & Schmolze, 1985). In addition to its representational role as a frame language, it is used to provide object-oriented definitions for scenario agents and entities. The rule language, one of the several versions of Flavors Expert (FLEX) (Shapiro, 1984), is a forward chaining rule language with collections of rules segregated into individual rule sets. The procedure language, the Simulation Core (SCORE) language, is used to define the behaviors of all scenario agents. It includes forms for defining proactive behaviors as goals and procedures, and reactive behaviors initiated by impinging events. Multiple task behavior is mediated via priority-based procedure conflict resolution.

A publish-subscribe protocol forms an important component of the SCORE language. *Signal-event* and *signal-event-external* are the basic forms for "publish;" the former broadcasts a message within a node, while the latter broadcasts a message locally and to remote nodes. The message is in the form of a list where the first element of the list is the message type. One or

more agents may subscribe to a signal by type in one or more procedures. The individual elements of the message may be vetted before deciding to accept the message for further processing. The subscription to a message type expires when a message is accepted for processing and must be renewed if further messages of that type are to be processed.

In a distributed simulation environment, simulator-to-simulator communication is required in addition to agent-to-agent communication. The form *signal-event-simulator-external* is provided to support this functionality. Simulator-to-simulator communication is used primarily for time management in distributed simulation.

Data recording forms an important subsystem within the Core-OMAR simulator. The SCORE language includes forms to support data recording. The objects recorded are event objects defined using the *defevent* form. Two built-in event types, *stimulus* events and *response* events, were used to capture the timing of the events that reflected human or model performance. Data collected reflected the timing of a response to a given stimulus or the failure to respond to the stimulus. Additional event types were defined to record scoring data associated with participant errors and timing penalties. At the end of a simulation run, data relevant to the run, including stimulus-response event data and scoring data were recorded to disk.

In addition to its role as a peer in a network of homogeneous or heterogeneous simulators, each Core-OMAR node can act as a server supporting several clients used to complete the simulation environment. The clients, all implemented in Java, provide user interfaces to support scenario development and execution. The developer's interface includes a graphical editor to support the definition of SFL concept and roles, and a browser to examine the detailed and large-scale structure of the goal and procedures that make up a scenario. The simulation control panel is used to select and control scenario execution.

The ATC Workplace (Figure 13) as operated by a human participant was implemented as an application interface. For the experiment trials, the simulation control panel enabled the experimenter to select and initiate the scenario to be executed.

3.2.3 D-OMAR Native-Mode Distributed Simulation

The AMBR Experiment 1 Phase 1 and Experiment 2 trials were run using D-OMAR native-mode connectivity between simulation nodes. For human participant trials, the D-OMAR simulator was run in real-time mode. For Experiment 1, the experimenter selected and initiated a scenario using the control panel and the human participant interacted with the ATC workplace as

required by the unfolding events. For Experiment 2, a scripting mechanism was developed to automatically sequence through the training scenarios and the subsequent trial scenarios. This significantly reduced the workload for the experimenters.

For model trials, the D-OMAR simulator was run in fast-time mode. System configurations for the four human performance models were each slightly different. The Experiment 1 DCOG model was developed as a D-OMAR model. As such, it operated as a D-OMAR agent in the same Lisp image as the experiment scenario. The Experiment 2 DCOG model was written in Java, operated in its own Java virtual machine, and communicated with D-OMAR through a socket.

The ACT-R and EASE models are Lisp models that each operated in its own simulator. Hence, each was configured as a two-node network. The ACT-R model was run in the same Lisp image as the D-OMAR simulator. The EASE model ran under Linux. It ran on a single machine with the Linux version of D-OMAR. In this instance, D-OMAR and EASE each operated in its own Lisp image connected via a socket.

Lastly, the COGNET/iGEN model, a C++ model, ran in its own simulator and used a CORBA interface to support connectivity with D-OMAR.

The varied distributed computing environments employed for the AMBR experiments made it essential that time-management and data exchange be carefully addressed.

3.2.3.1 Native-Mode Time Management

For the Experiment 1 DCOG model, time management was not an issue; it simply ran within the D-OMAR simulator alongside the standard AMBR scenario entities. Each of the other three models ran in its own simulator interacting with the D-OMAR simulator. From the perspective of the D-OMAR simulator, the basic time management cycle was to complete an event-based time-step in which an update of the ATC workplace took place and then grant the model's simulator the opportunity to respond to the new screen events and act on any of its pending initiatives. The grant included notification of the time at which the D-OMAR simulator required control again. The model simulator was free to run up to the grant time or to an earlier time at which it generated an input to the workplace. At this point, the model's simulator would issue a symmetrical grant specifying the current time and the time at which it next needed control. With the grant now passed back to D-OMAR, the basic time cycle then repeated once again. The pattern continued until reaching the stop time dictated by the scenario for the trial.

In addition to the basic time grant, the API included start-of-run and end-of-run notifications. On the D-OMAR side, the API was implemented using the standard publish-subscribe protocol with the *signal-event-simulator-external* form used to generate outbound messages and the *with-signal* form used to capture and process inbound messages. The ACT-R and EASE teams using Lisp and the DCOG team using Java were provided code to support these exchanges. D-OMAR and COGNET/iGEN handled these exchanges using CORBA.

3.2.3.2 Native-Mode Data Exchange

Data exchange between D-OMAR and each of the HBR models included: (1) information that was presented at the workplace to be viewed by the model, and (2) actions that the model could take to construct messages to adjacent controllers and to aircraft in the sector. D-OMAR provided information on *what* was presented on the screen; it was the responsibility of the model simulator to determine *how* the model "saw" the data. D-OMAR provided the models with updates of aircraft positions and, in the color case, updates of the color changes that specified when actions were pending on an aircraft.

Actions taken by the models emulated the mouse object selects used to construct messages to adjacent controllers and sector aircraft. The messages included a message type and the slot values necessary to complete the message. Some message types required a single argument, an aircraft select, others required an aircraft select and an ATC select. A mouse select of the "Send" button initiated the transmission of the message. As messages were constructed, the model was notified of updates to the prompt panel above the radar screen, first with the template for the message type, and then with the slot values are as they were entered. When a message was transmitted, the model was notified of the clearing of the prompt panel and of the message's appearance as an outgoing message in the appropriate message panel. The models were also notified of the appearance of inbound messages from adjacent controllers and sector aircraft and the message panel in which they were to appear.

The API for data exchange was implemented much like that for time management. The *signal-event-external* form was used to generate the messages notifying the model of new or updated information appearing at the ATC workplace. Messages arriving from the models that detailed mouse selects were captured and processed using the *with-signal* form.

3.2.4 HLA-Mode Distributed Simulation

For Experiment 1 Phase 1, the central focus was the conduct of the multitasking experiment itself. For Experiment 1 Phase 2, the focus was on replicating the Phase 1 results in the High Level architecture (HLA) (Kuhl, Weatherly, & Dahmann, 1999) simulation environment. Earlier work in D-OMAR included an HLA interface for real-time execution using HLA interactions for data exchange. For the AMBR experiments, the D-OMAR interface to HLA was upgraded to also address fast-time time-management and the attribute-value model for data exchange. MITRE (Feinerman, Prochnow, & King, 2001) provided expert advice in the development of the HLA federates. DMSO release 1.3 NG V3.2 of the RTI was used for the AMBR HLA-based experiment.

Implicit in the design for the HLA experiment was the separation of the workplace from the entities reflected in that workplace—the aircraft and the air traffic controllers for the four adjacent sectors. In moving to the HLA implementation, two federates were developed. The first federate, the "workplace" federate, provided the ATC workplace to be operated either by a human participant or by a human performance model. The second federate, the "world" federate, included models for the aircraft transiting the airspace and the human performance models for the controllers for the adjacent sectors.

When running with a human participant, the simulation environment included just two federates (not including the HLA federation tools), the world federate and the workplace federate operated by the human participant. When running with a human performance model, the model operated as an agent in a third federate interacting solely with the workplace federate.

3.2.4.1 HLA-Mode Time Management

Time management in an HLA federation is handled by the HLA Run Time Infrastructure (RTI). Federates send requests to the RTI when the federate is ready to advance their local clocks, either to a specified time, or to the time of the next incoming event. The RTI is responsible for keeping all federates synchronized by appropriately granting time advances.

When running in fast-time with a human performance model, the world and workplace federates used the RTI's *next-event-request* (NER) command with a time-out value in order to advance their clock. The D-OMAR native-mode time-grant message mapped directly onto the RTI *next-event-request* command. The RTI allows the federate to advance its logical clock either to the time of the specified request or to the time at which a new event for the federate was

generated, whichever is earlier. When a federate advances its logical clock, the next possible time at which it can generate an event is equal to the time of the federate's *time-advance-request* plus the federate's *lookahead* value. When a human performance model hits the Send button, the workplace federate is notified immediately and must then immediately notify the model's federate of the updated screen state. We used a *lookahead* value of zero, so that a federate could immediately generate events at the time to which it had advanced. Larger *lookahead* values facilitate increased parallel processing by the federates in an HLA federation. The requirements of the experiment prevented taking advantage of this HLA capability.

For the human participant trials, no formal RTI time management protocol was necessary, since the human interaction must occur in real time. In this case, the RTI acted as a message passing system, forwarding events to other federates as quickly as possible. Since there is no mechanism to guarantee hard real time performance, the hardware was selected to be fast enough to ensure that all messages were processed in a timely manner.

3.2.4.2 HLA-Mode Data Exchange

In order to handle data exchanges in the HLA federation, we created a one-to-one mapping from the D-OMAR native mode API to HLA exchanges using a mix of *interaction* and *attribute-value* updates. We created HLA objects to represent the aircraft and ATC regions. Updates of aircraft positions were transmitted from the world simulation federate to the workstation federate and from there to the human performance model federate through the HLA attribute-value update function. Data exchanges such as communication messages from an ATC to an airplane, or button presses at the ATC workplace either by a human participant or human performance model, were transmitted as HLA interactions.

Additional interactions were needed for the HLA federation since the world-simulation and the workplace were implemented as two separate federates. In addition to the standard aircraft position-change messages, the world-simulation federate had to inform the workstation whenever an aircraft crossed the notification boundary. This allowed the workstation to accurately record the time of the boundary crossing as a stimulus event and thus pair it with the appropriate subsequent button-press response event.

3.2.4.3 HLA Impact on Model Performance

An important goal for Experiment 1 Phase 2 was to demonstrate that the HLA provided a simulation environment in which it was reasonable to conduct human performance experiments

and that human performance models could reasonably interact with a simulated workplace. The HLA implementation met these goals, but was found to have a significant negative impact on the run-times for most of the human performance model trials. The HLA configuration was also found to require significantly more computer power for the human participant trials.

For the HLA human participant trials, additional computer power was required to assure adequate real-time system response. For Phase 1, the experiment control panel and the simulation were run on a 200 MHz Pentium desktop machine. The workplace display operated on a 500 MHz Pentium laptop machine. For Phase 2, the HLA implementation required the replacement of the 200 MHz machine with a 266 MHz machine and the addition of a second 500 MHz machine. All three machines had 128 Mbytes of memory and operated under Windows NT. The world federate ran on the 266 MHz Pentium laptop, the workplace federate and the HLA RTI ran on a 500 MHz Pentium laptop, and the workplace display ran on the second 500 MHz Pentium laptop. The experiment was controlled from the laptop running the workplace federate. For Experiment 2, the machine configuration was the same as Experiment 1 Phase 1, with the 200 MHz desktop machine replaced by a 500 MHz laptop machine.

For the model runs, there was a price to pay for the distributed computation. Table 6 provides data on run times relative to real-time (*e.g.*, the DCOG model ran more than 14 times faster than real-time in native-mode and two-thirds real-time in HLA-mode). The data in Table 6 reflects DCOG, ACT-R, and COGNET/iGEN trials that were run on 950 MHz Pentium with 512 Mbytes of memory operating under Windows NT. The Phase 1 DCOG, ACT-R, and EASE trials used native-mode D-OMAR connectivity. The Phase 1 COGNET/iGEN trials used a CORBA interface between the model and the D-OMAR simulator. As indicated in Table 6, the HLA RTI interface was actually more efficient than the CORBA interface. It was the one case in which HLA provided improved performance over Phase 1.

The EASE trials were run on a 400 MHz Pentium with 256 Mbytes of memory operating under Linux, hence the EASE timing data is not directly comparable with the data for the other models. The relative speeds of native-mode and HLA-modes for the EASE runs are relevant—HLA was slower by a factor of two.

Table 6: Run-time as a Multiple of Real-time in Native-Mode and HLA-Mode AMBR Trials

	950 MHz 512 Mbytes NT			***400 MHz 256 Mbytes Linux***
	AFRL DCOG	*CMU ACT-R*	*CHI Sys COGNET*	*Soar Tech EASE*
Native Mode	0.07	0.11	1.30	2.27
HLA Mode	1.48	0.86	0.94	5.55

The human performance models that ran very fast (see Table 6) in native-mode lost this advantage in HLA-mode. The dominant performance factor was not in the models themselves, but rather the mode of connectivity between the model's simulator and the D-OMAR simulator. Socket connections, the HLA RTI, and particularly the CORBA implementation each had a high cost associated with their use.

A second factor impacting performance was the necessary implementation of the scenarios as two federates, a world federate and a workplace federate, for the HLA implementation. It was important to demonstrate a generic workplace readily adaptable to operate with a broad range of vehicle federates, but this did have an unfavorable impact on performance. Message traffic that was local to a single simulator in Phase 1 became message traffic moving between two federates in the HLA implementation.

Operating in D-OMAR native-mode, the DCOG and ACT-R model trials were dramatically faster than real-time. This is clearly the regime in which one would prefer to operate. Short run-times facilitated model development and model trials by compressing AMBR experiment trials with wall clock times of six and a half, nine, and eleven and a half minutes to a minute or less. The Phase 1 DCOG model ran internal to D-OMAR and did not require a socket connection. The ACT-R model ran in the same Lisp image as D-OMAR. Native-mode D-OMAR code recognized the shared image and bypassed the socket connection required between Lisp images or between nodes. This optimization led to very significant time savings.

EASE ran in a separate Lisp image and required a socket connection to D-OMAR for non-HLA trials. The COGNET/iGEN model used CORBA to connect to D-OMAR for non-HLA trials. The necessary socket connection significantly slowed execution times for these model runs.

The results of the Phase 1 trials were reproduced in the HLA environment, but were found to require significantly more computer power to maintain adequate system response for the human participant trials. For three of the four human performance models, trial run-times were significantly longer in HLA-mode than in D-OMAR native-mode. HLA provides another time management mechanism, the time advance request (TAR). It is possible that TAR would have proven more efficient, but there were not time or resources available to explore this option.

When computer resource utilization is a concern in an HLA simulation environment, consideration should be given to implementing the workplace and the human performance model as a single federate. The efficiency of within-simulator communication can be used to efficiently accommodate the demand for high frequency data exchange between the workplace and the human performance model.

3.2.4.4 HLA Federate Compliance Testing

The world-simulation federate and the workstation federate both completed the DMSO-sponsored HLA compliance testing on March 22, 2001. This certifies that the two federates are fully compliant with HLA version 1.3, and makes them available to other researchers. The world simulation federate is quite domain specific, useful only for simulating aircraft and ATCs in this simplified air traffic control environment. The workstation, however, is more generic. The workplace displays and functionality have been used in scenarios unrelated to the AMBR project and could readily be extended to operate in related domains. The federates are included on the CD that accompanies this book.

3.2.5 Conclusion

The computing environment to support the AMBR experiments was necessarily complex. It had first to support the implementation of the design for the two experiments; it also had to provide a real-time simulation environment in which to conduct the human participant trials and a fast-time simulation environment to support the model trials. Data collection and scoring had to operate identically for human participants and models as subjects. Connectivity with the simulators for each of the human performance models was slightly different in each instance. For Experiment 1 Phase 2, it was necessary to provide an HLA simulation environment for human participant and model trials. For the most part, we were able to rely on existing D-OMAR simulation capabilities to meet the demands of the AMBR experiment. It was sufficient to use existing capabilities to implement the scenarios required by the experiment designs, make modest improvements to the

existing HLA simulation capability, and provide scripting to support the Experiment 2 human participant trials and ease the burden of doing large numbers of human performance model runs.

3.2.6 Acknowledgement

The authors wish to acknowledge the assistance of David Prochnow and Ron King from The MITRE Corporation in developing HLA simulation environment for Experiment 1.

3.2.7 References

Brachman, R. J. & Schmolze, J. G. (1985). An overview of the KL-ONE knowledge representation system. Cognitive Science, 9, 171-216.

Deutsch, S., & Benyo, B. (2001). The D-OMAR simulation environment for the AMBR experiments. *Proceedings of the Tenth Conference on Computer Generated Forces and Behavior Representation.* Orlando, FL.

Deutsch, S. E. & Cramer, N. L. (1998). OMAR human performance modeling in a decision support experiment. *Proceedings the 42nd Annual Meeting of the Human Factors and Ergonomics Society,* Chicago, IL.

Feinerman, L. E., Prochnow, D. L., & King, R. A. (2001). Icarus: An HLA federation for HBR models. *Proceedings of the Tenth Conference on Computer Generated Forces and Behavior Representation.* Orlando, FL.

Kuhl, F., Weatherly, R., & Dahmann, J., (1999). *Creating Computer Simulation Systems: An Introduction to the High Level Architecture,* Upper Saddle River, NJ: Prentice Hall.

MacMillan, J., Deutsch, S. E., & Young, M. J. (1997). A comparison of alternatives for automated decision support in a multi-tasking environment. *Proceedings of the Human Factors and Ergonomic Society 41st Annual Meeting,* Albuquerque, NM.

Shapiro, R. (1984). *FLEX: A tool for rule-based programming.* BBN Report No. 5843. Cambridge, MA: BBN Technologies.

4. Comparison, Convergence, and Divergence in Models of Multi-tasking and Category Learning and in the Architectures Used to Create Them

(David E. Diller, Kevin A. Gluck, Yvette J. Tenney, Katherine Godfrey)

This chapter marks the beginning of the final section of the book (see Gluck and Pew, in press), in which we develop our conclusions, describe our lessons learned, and define some of the implications for research. This particular chapter does assume some familiarity with the material that has preceded it, and we recommend the reader refer back to those earlier chapters as necessary. Chapter 2 (in this report), for instance, describes the air traffic control task, the experiment designs, and the human data in detail, and we do not repeat those details here. Similarly, the preceding four chapters (see Chapters 4-7, Gluck and Pew, in press) provided detailed descriptions of the multitasking and category learning models developed by each of the modeling teams. The model description chapters were long and thorough by design, to allow the modelers the opportunity to provide a complete account, in unusual depth, of their architectures and of their modeling approach and implementation. This chapter is designed to provide a side-by-side comparative view of the models across a number of different dimensions.

We start with the models' ability to fit the observed human data – providing a comparative quantitative evaluation of model performance. We illustrate places where the models produce results similar to one another, as well as where they make their own unique predictions. It is these similarities and differences that help us better understand the processes by which we as humans operate effectively in complex tasks and also contributes to our understanding of the kinds of representations and processes that make such behaviors possible in computational models. We follow the comparison of model fits to human data with a discussion of other dimensions along which one might compare computational process models and some of the challenges associated with comparing along those dimensions. These dimensions include the degrees of freedom available in the architectures and in specific model implementations, model reuse, interpretability, and generalizability.

From a focus on model comparison, we turn to a discussion of the similarities and differences among the modeling architectures. The authors of the model description chapters each addressed a set of common questions that were considered to be of broad theoretical and practical significance for those interested in the science of human representation. We draw from our

experience with the models and the answers to the common questions to present a summary in both narrative and table formats.

We conclude the chapter with a discussion of points of convergence and divergence in the models of multitasking and category learning developed for the AMBR Comparison, and in the architectures used to create them.

4.1 Quantitative Fits to the Experimental Results

4.1.1 Experiment 1: Air Traffic Control Procedure

COGNET/iGEN, DCOG, and EASE provided model runs that produced data equivalent to that of four participants. ACT-R provided runs equivalent to sixteen participants, to match the number of human participants in the study. The number of runs was determined by time considerations and modelers' preferences.

Model predictions are compared against the observed human data using either a sum of the squared error (SSE) or a G^2 measure. SSE was used for continuous variables such as reaction time, while G^2, sometimes known as deviance, was used for categorical or counted data, such as accuracy results. G^2 is a log-likelihood ratio statistic designed to measure the goodness of fit between predicted and observed data and, like χ^2, is calculated for contingency tables. See Agresti (2002) for additional details on categorical data analysis.

4.1.1.1 Accuracy as a Function of Display and Workload

Figure 14 illustrates both human data and each model's predictions of mean accumulated penalty points by condition. Error bars represent dual-sided 95% standard error of the mean confidence intervals.

All four models correctly produced higher penalty points in the text display than in the color display, especially at higher workloads. Color display conditions produced very few penalty points in both the observed and predicted data. With the exception of DCOG, models tended to over predict the number of penalty points in the text display condition. DCOG under predicted the number of penalty points in the text display condition. COGNET/iGEN came closest to the human norms with an SSE of 1,745 followed by ACT-R with 15,752, DCOG with 29,151 and 150,034 for EASE.

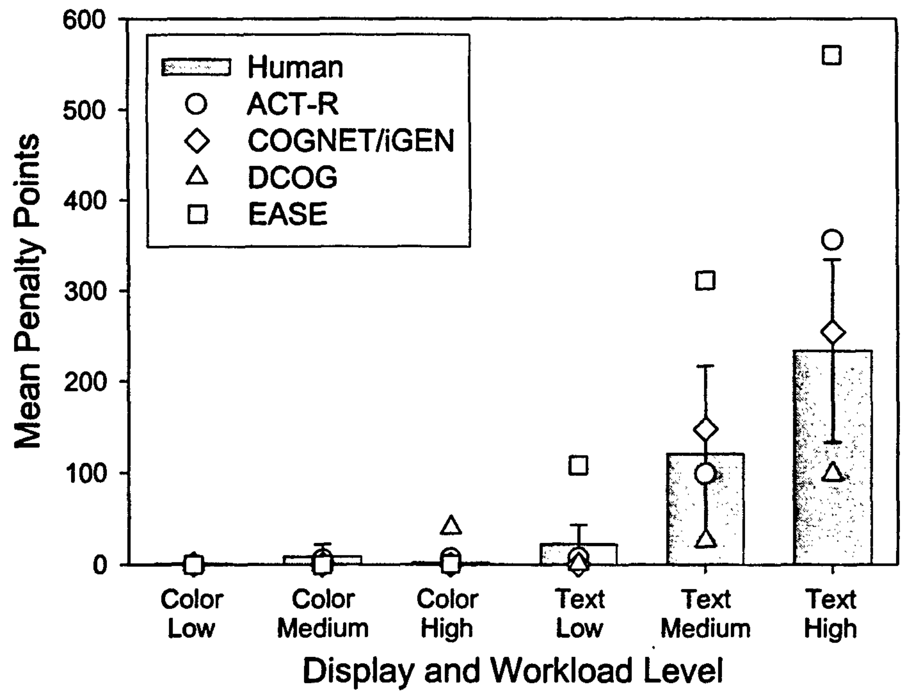


Figure 14. Human and model penalty scores as a function of display and workload.

Penalty scores were explored in greater detail in the most demanding condition: text display with high workload. The upper panel in Figure 15 shows the penalty points earned by humans and models in each of the penalty subcategories for the text-high workload condition. It is clear from the graph that the overriding source of points for humans was Hold penalties (at 50 points each). All the models showed this same pattern, although no model fell within the human confidence intervals for each penalty type. Again, deviations from the observed data tended to be in the direction of too many penalties. DCOG came closest to the human norms, with an SSE score of 35,193, followed by 60,545 for COGNET/iGEN, 94,927 for ACT-R, and 234,665 for EASE.

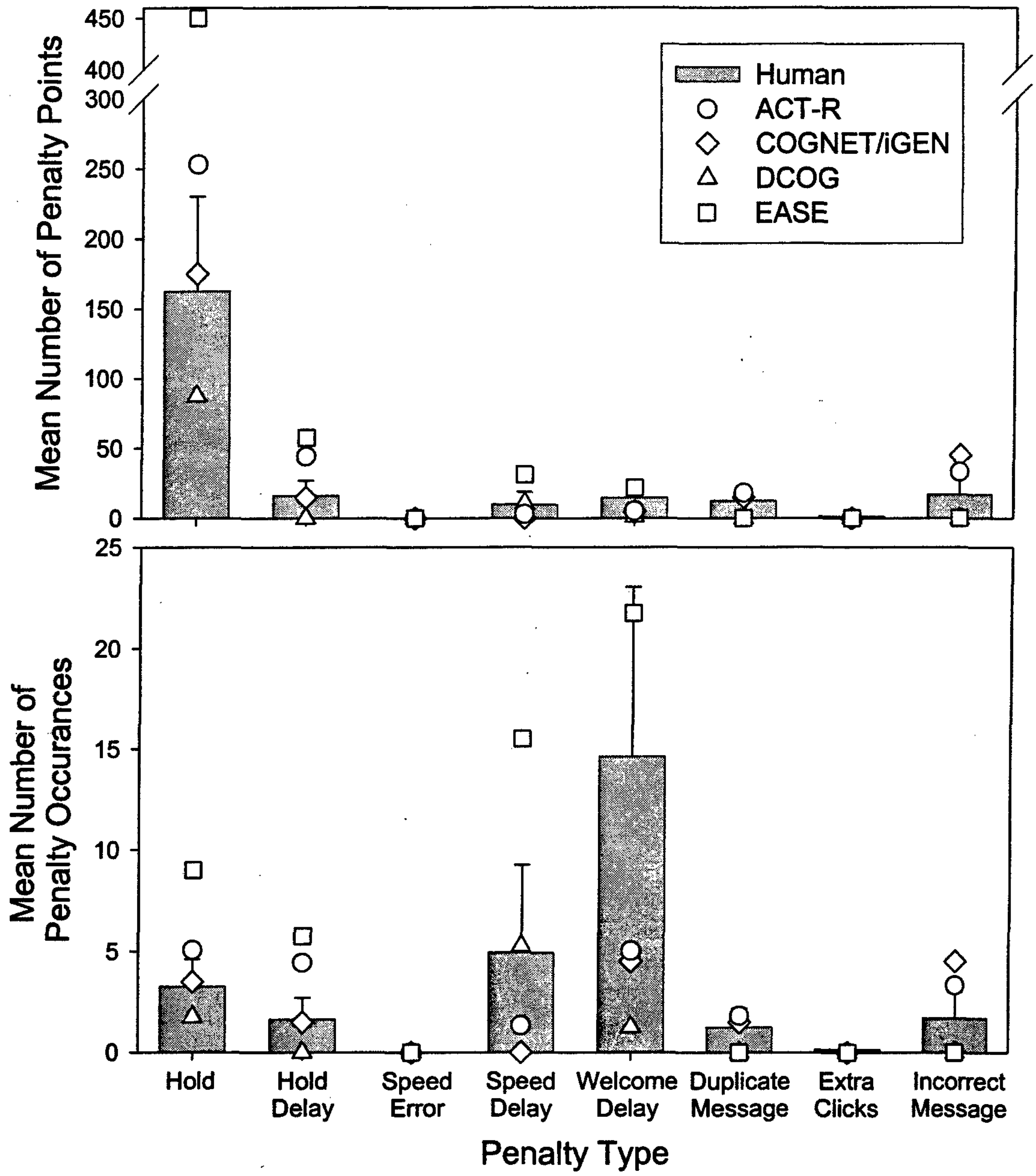


Figure 15. Human and model performance by penalty category for text-high workload condition.

The lower panel in Figure 15 shows the actual number of occurrences of each type of error. The results suggest that participants prioritized their actions so as to minimize overall penalties. Thus, Welcome Delay, which carries the lowest penalty (1 point per minute), was the most frequent penalty obtained by humans. The next largest category of observed errors was Speed Delay (2 points per unit of time). The "load shedding" strategy of postponing actions carrying low

penalties to focus on preventing aircraft from turning red, which carries a higher penalty (50 points), is a reasonable strategy for coping with high workloads. None of the models managed to consistently fall within the confidence limits of the observed data. However, EASE did show evidence of load shedding. EASE was the only model to have more occurrences of Welcome Delays and Speed Delays than of Holds, resembling the observed data. SSE scores were 288 for DCOG, 313 for COGNET/iGEN, 342 for ACT-R, and 1,083 for EASE. Interestingly, despite being the only model to show load shedding behavior, EASE has the worst SSE value. Clearly, the type of measures, such as relative trend or quantitative measures, used to evaluate a model can greatly impact conclusions about the quality of the model. Schunn and Wallach (2001), make the point that it is possible to have an inverse relationship between qualitative trends and absolute fit measures and like EASE fit the trend, but provide a poor fit to the absolute data, or in contrast provide a reasonable fit to the absolute data, but miss the trend, illustrating the need to evaluate both relative trend and absolute deviation from the data.

4.1.1.2 Response Time as a Function of Display and Workload

Figure 16 illustrates the human and model response times for each condition. As can be seen in the graph, participants responded to the events more quickly with the color display than with the text display, and workload effects were more pronounced in the text than in the color condition. These results show a similar pattern to the results seen in accuracy measures, suggesting there was no speed/accuracy tradeoff occurring for the conditions. The models all showed similar trends. No model fell within the confidence intervals for all conditions, but again COGNET/iGEN came extremely close. SSE scores were 6 for COGNET/iGEN, 80 for ACT-R, 276 for DCOG, and 285 for EASE. Overall, the models tended to respond too slowly, relative to the observed human data.

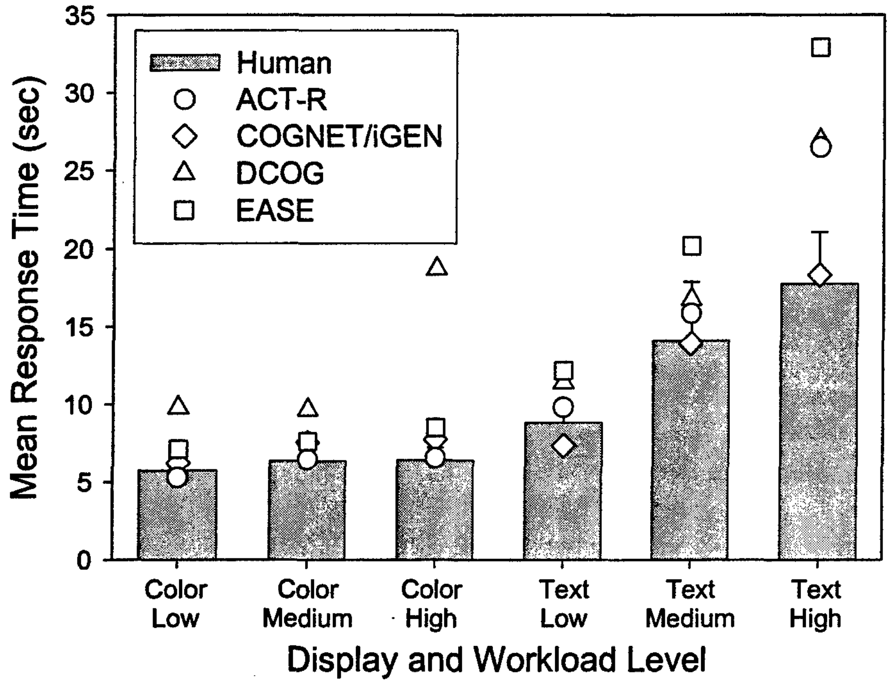


Figure 16. Human and model mean response times as a function of display and workload.

4.1.1.3 Subjective Workload Measures

Human subjects provided separate data for each of the six individual workload scales (mental demand, physical demand, temporal demand, performance, effort, frustration) that are part of the Task Loading Index (TLX) workload rating sheet (Hart & Staveland, 1988; Vidulich & Tsang, 1986). Each individual scale rated workload from 0 to 10 representing low to high workload, respectively. COGNET/iGEN produced a workload score for each of the six TLX scales. ACT-R and EASE each produced a single overall workload score. DCOG did not calculate workloads. To allow comparison of the ACT-R, EASE, and COGNET/iGEN workload ratings, an overall subjective workload rating was obtained for each human subject by averaging the six scores on the individual TLX scales. The COGNET/iGEN model values were similarly averaged across the six scales.

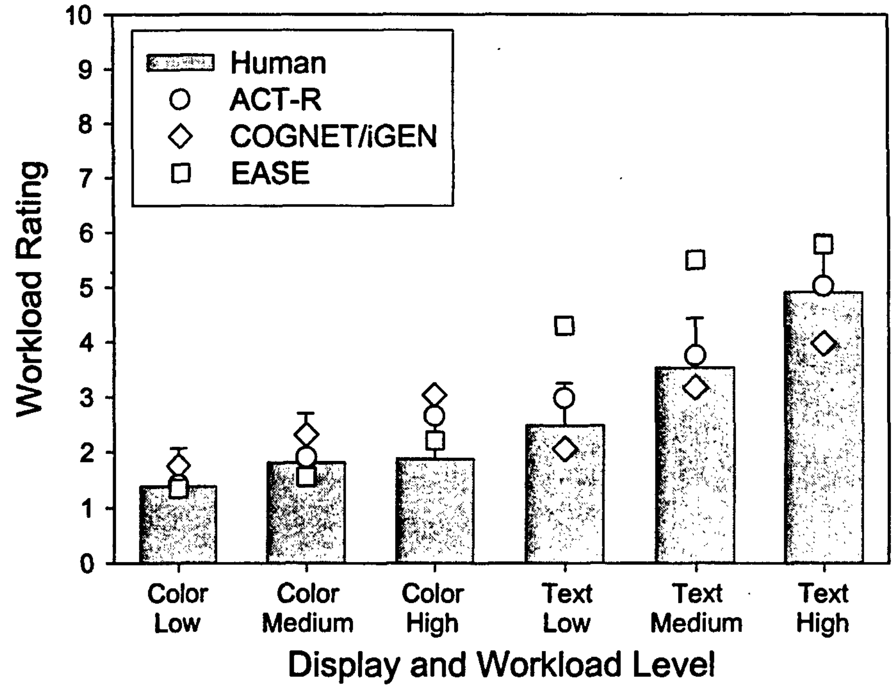


Figure 17. Human and model subjective workload as a function of display and workload condition.

The aggregate human results, shown in Figure 17, demonstrate that participants rated their workload as higher for the text than for the color display. There was also an increase in subjective workload as actual workload increased, especially for the text display. The three models showed the same pattern of results and provided a good qualitative fit to the data. ACT-R provided the best fit with an SSE of 0.92 and falling within the observed confidence intervals for all conditions. COGNET/iGEN and EASE produced SSEs of 2.91 and 8.06, respectively.

4.1.1.4 Discussion

The assortment of quantitative data reviewed above provided a significant set of challenges to the models. All the models were successful in producing the qualitative effects of display type and workload on penalty scores. With respect to reaction time measures, all models showed the general trend of reaction times increasing with workload level. In addition, all models except DCOG produced average response times in the color display conditions that were faster than the easiest text display condition. DCOG failed to produce this result because its model's performance was affected by workload level in the "Color-High" condition. Although a main focus of this experiment was on multiple task management, only one of the models, EASE,

showed evidence of load shedding, with more occurrences of Welcome Delays and Speed Delays than of Holds, resembling the observed data. Ironically, despite being the only model to get this particular qualitative result correct, EASE had the worst quantitative fit to the data on penalty points and penalty occurrences. All the models produced the qualitative relationship between subjective workload rating and workload level, except DCOG, which produced no workload ratings. In general, while most qualitative trends were produced, close quantitative fits were achieved only infrequently.

4.1.2 Experiment 2: Category Learning

The category learning experiment, which involved a modification to the air traffic control task used for Experiment 1, is described in detail in Chapter 2 (in this report) and we will not repeat those details here. Data were collected from 90 participants in that study, and each modeling team completed enough runs to simulate the 90 participants. Three of the models produced subject variation through stochastic variation of parameters on each run (COGNET/iGEN, ACT-R, EASE). One developed "templates" of a smaller number of subjects, defined by parameter values or strategy choice, and then replicated them to produce the requisite number of subjects (DCOG). All the models completed the main task, workload ratings, and transfer test (but not the training blocks, quiz, or debrief). One model, COGNET/iGEN, produced ratings for the six NASA TLX workload scales used by the human subjects, rather than a single composite workload score.

All human results, with the exception of the transfer task, were shared with the modeling teams early in the model development cycle to facilitate their modeling efforts. The results of the transfer task were not revealed to the modeling teams until after the modeling teams produced the initial round of model predictions. This manipulation was meant as a test of the model's ability to predict, and not simply replicate, human behavior. The results from this initial round of model prediction were compared to human performance. Modeling teams were then provided with the results of the transfer task and allowed to revise their models in light of these results. During this round of model revisions, the EASE modeling team introduced a second variant of the EASE model. This new variant, called RULEX-EM, was derived from the Rule-Plus-Exception (RULEX) model developed by Nosofsky et al. (1994a). In addition, the EASE team revised their original model, based on the Symbolic Concept Acquisition (SCA) model (Miller & Laird, 1996). All of the models were evaluated against the observed data a second time, to determine

whether they had been successful in creating a better explanation of human performance on the transfer task.

Because the models were tasked to generate 90 simulated participants, we decided to analyze and compare the models against the pattern of main effects and interactions seen in the analysis of variance on the human data. Each of the 90 model runs was treated as an independent participant and analyses of variance (ANOVAs) were generated in the same manner as for the human participants. Results of these ANOVAs for both human and models are presented in table format for each dependent variable evaluated.

4.1.2.1 Category Learning Task (Primary Task)

Accuracy measures. The observed category learning data and the data from each of the models are shown in Figure 18. We plot the mean probability of error for each block of 16 categorization judgments for both human and models in each of the Type I, III, and VI problems. Each model's initial and revised predictions and the observed human data are organized into a row of three graph panels, one panel for each problem type.

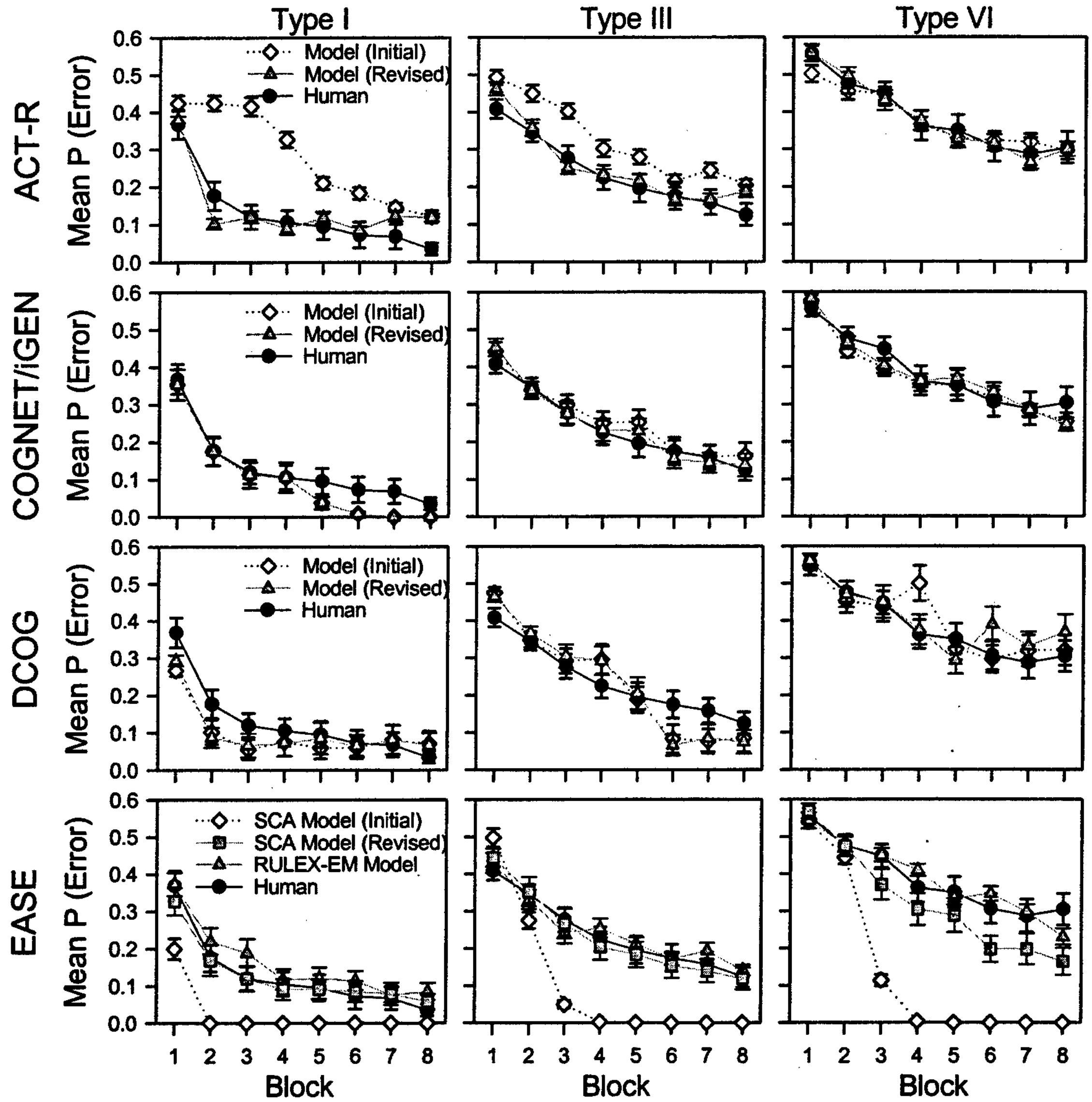


Figure 18. Human category learning data for Type I, III, and VI problems and initial and revised model data.

G^2 analysis of the initial model fits to the Type I, III, and VI data were 15.95 for DCOG, 21.36 for COGNET/iGEN, 49.69 for ACT-R, and 673.62 for the EASE SCA model. Additionally, the models were evaluated using the same analysis of variance measures used to evaluate the human data in order to evaluate how well the models captured the observed pattern of results. These results are shown in Table 7. The first row describes the significant (or non-significant, as the case may be) main effects and interactions in the human data. Check marks indicate those instances where the model replicated the observed human results. A significance level criterion of .05 was used for rejecting the null hypothesis. The initial ACT-R model showed the desired

main effect of problem type and main effect of block, but also showed an interaction of Problem Type x Block that was not observed in the human data. In particular, the shape of the learning curve, especially for Problem Type I, tended to drop too slowly during the initial blocks, in contrast to the rapid learning seen in the observed data. The initial COGNET/iGEN model showed significant effects of problem type and block, with no interaction of Problem Type x Block. As can be seen in Figure 18, the COGNET/iGEN model matched the observed data well, falling within, or very close to, the observed Standard Error of the Mean (SEM) in all cases except the later blocks for Problem Type I. The initial DCOG model showed the desired effects of problem type and of block, but showed a Problem Type x Block interaction not present in the human data. The initial version of EASE, the SCA model, showed a significant problem type and block effect. However, the SCA model differed significantly from the observed human data in several respects. EASE exhibited an inappropriate interaction of problem type x block. As can be seen, there was a precipitous drop in errors to zero, especially in Problem Type III and VI, as compared to the more gradual learning curve in the observed data.

Table 7. A Comparison of Human and Model Data for Primary Task Accuracy Measures.

	ANOVA Main Effects and Interactions		
	Problem Type	*Block*	*Prob Type by Block*
Human	***Significant****	***Significant****	***Not Significant**+*
Original Model Predictions			
ACT-R	✓	✓	
COGNET/IGEN	✓	✓	✓
DCOG	✓	✓	
EASE SCA	✓	✓	
Revised Model Predictions			
ACT-R	✓	✓	
COGNET/IGEN	✓	✓	✓
DCOG	✓	✓	
EASE SCA	✓	✓	
EASE RULEX-EM	✓	✓	

* $p < .0001$, + $p > 0.05$

After a round of model revisions, all of the models showed improvement in their quantitative fit to the human data, some of them dramatic improvements. The ACT-R model showed a marked improvement, lowering its G^2 value from 49.69 to 7.23. The revised ACT-R model again showed significant effects of problem type and block, but still had an inappropriate interaction of Problem Type x Block. The final COGNET/iGEN model showed an improved fit to the observed data, with a G^2 of 20.92, the appropriate main effects, and no interaction. The fit of the DCOG model showed a slightly improved G^2 value of 15.53. There was still an inappropriate significant interaction of Problem Type x Block. As can be observed in the DCOG Problem Type III and VI plots in Figure 18 the learning curves were more irregularly shaped than the observed data. The EASE SCA model showed a dramatically improved fit, lowering its G^2 value to 9.96. However, the EASE SCA model still showed the interaction of Problem Type x Block. The EASE RULEX-EM model showed a G^2 fit of 5.64, with main effects of problem type and block, but also the unobserved interaction of Problem Type x Block.

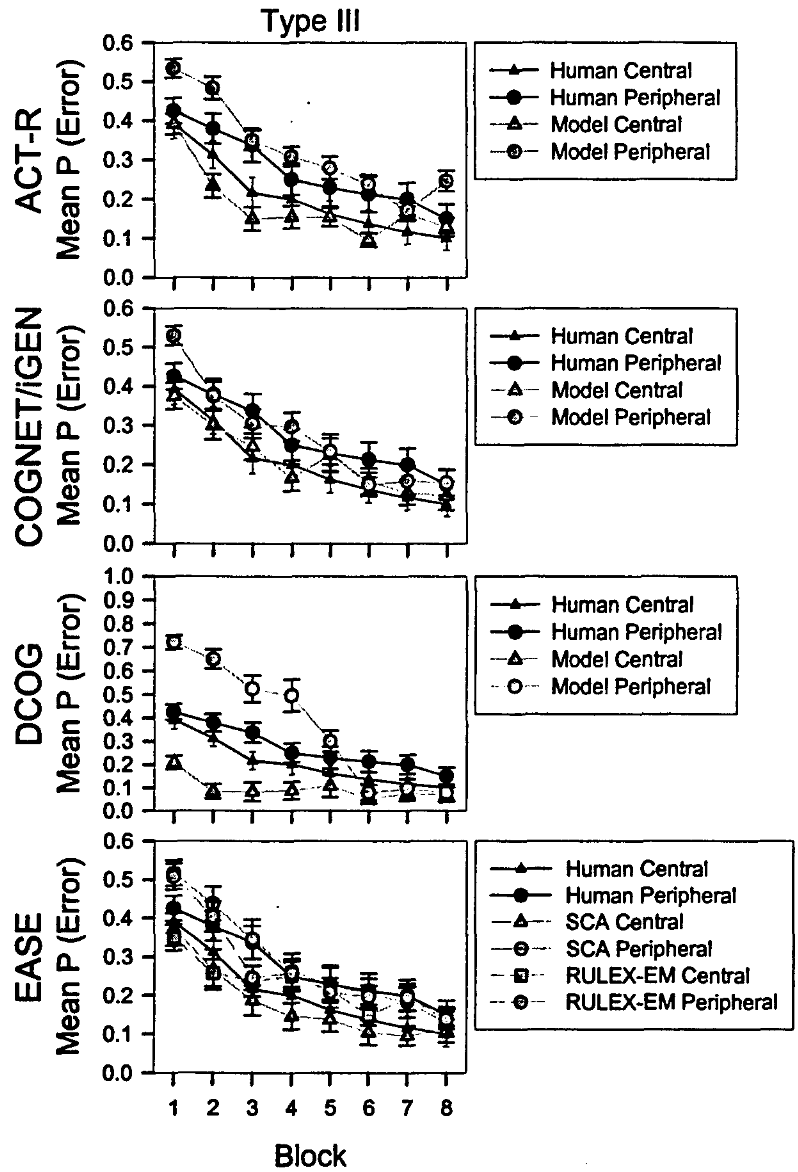


Figure 19. Human and revised model data for the Type III problem learning data.

A more fine-grained analysis of Problem Type III is shown in Figure 19. In Problem Type III, half the stimuli are members of the "central" set and half are members of the "peripheral" set.[4] The results show humans learned central stimuli more quickly than peripheral stimuli. The predictions of the revised models for central and peripheral item types are shown in Figure 19. It is clear from the graph and the item effect from Table 8 that all models except EASE RULEX-EM made fewer errors in learning the central than the peripheral items. G^2 results for the revised models were 3.46 for EASE SCA, 4.40 for COGNET/iGEN, 5.89 for EASE RULEX-EM, 9.24

for ACT-R, and 74.49 for DCOG. A separate analysis of variance was conducted for each model alone, to determine if the model replicated the human results of significant effects of item (peripheral or central) and block, with no interactions between the two variables. However, none of the models matched the observed data perfectly, except for EASE SCA, which showed significant item and block effects, and no significant interaction between the two variables. The ACT-R, COGNET/iGEN, and DCOG models all showed significant effects of item and block, but also inappropriate interactions of Item x Block. EASE RULEX-EM, while showing an appropriate effect of block, failed to show a significant difference in learning rates for central and peripheral items and showed an inappropriate interaction of Item x Block.

Table 8: Revised Model Results for Central/Peripheral Item Differences

	ANOVA Main Effects and Interactions		
	Item	*Block*	*Item by Block*
Human	**Significant***	**Significant****	**Not Significant+**
Revised Model Predictions			
ACT-R	✓	✓	
COGNET/IGEN	✓	✓	
DCOG	✓	✓	
EASE SCA	✓	✓	✓
EASE RULEX-EM		✓	

* $p < .05$, ** $p < .0001$, + $p > 0.05$

Response time measures. Figure 20 shows the mean response times to the primary category learning task for both observed data and each model's revised fit. Response times to the category learning task were faster for Problem Type I than Problem Type III or VI, which did not differ from one another. Models showed a large variation in quality of fit. ACT-R response times were too slow, producing an SSE score of 30.72. ACT-R showed the desired effect of block, but not problem type. COGNET/iGEN had the closest fit with a SSE of 3.21 and showed the desired block effect. However they failed to show a problem type effect and had an inappropriate interaction of Problem Type x Block. The DCOG model had a SSE of 170.02 reflecting the fact that the response times were too slow. DCOG failed to achieve faster responses for Problem Type I than for Problem Type III or VI. EASE SCA had a SSE of 39.33, reflecting the fact that

[4] The central/peripheral distinction is explained in Chapter 2.

the response times were too fast for Problem Types III and VI. EASE RULEX-EM had a SSE of 8.40 reflecting a better fit to response times for Problem types III and VI. EASE RULEX-EM and EASE SCA both showed the desired block effect, but failed to achieve the problem type effect.

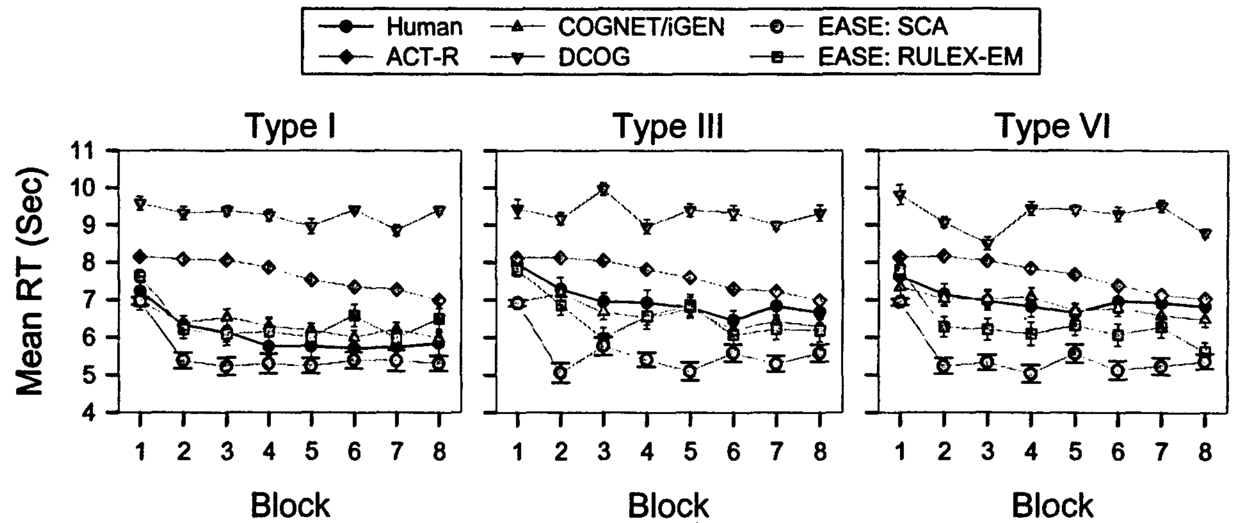


Figure 20. Human and revised model response times on the category learning task as a function of category learning problem type.

4.1.2.2 Handoff Task (Secondary Task)

Penalty score measures. Figure 21 shows the mean penalty score on the secondary task for Problem Type I, III, and VI for the observed data and revised model predictions. As described in Chapter 2 (this report), there were no effects of blocks or problem type. As illustrated by the error bars in the figure, the human data are quite variable. Revised quantitative model fits to the penalty score data are as follows (in SSEs): 1726.86 for COGNET/iGEN, 1924.06 for ACT-R, 2043.46 for EASE RULEX-EM, 2720.29 for EASE SCA, and 5098.42 for DCOG. In general, the models fit the qualitative data reasonably well. The DCOG model showed a number of blocks with too many penalty points, reflected in an undesirable significant main effect of block, $F(7,399) = 2.32, p = .0248$ and an inappropriate interaction of Block x Problem Type, $F = 4.52, p < .0001$. All other models accurately predicted no significant main effects or interactions. ACT-R, COGNET/iGEN and EASE SCA models showed scores that were close to the observed data, while EASE RULEX-EM showed too few penalty points.

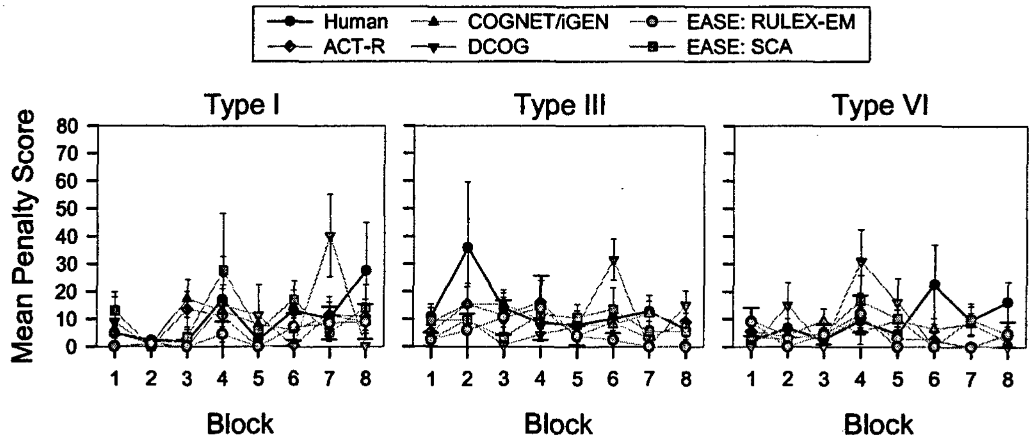


Figure 21. Human and revised model penalty scores on the handoff task as a function of category learning problem type.

Response time measures. Human and revised model response times for the secondary task are shown in Figure 22. Human response times were quite variable, illustrated by the large error bars in the graph. While no main effects of workload level or problem type were found, there was a main effect of blocks, with participants responding more quickly on later blocks. However, this appears to be primarily driven by the results observed in Problem Type III. There was no interaction of Problem Type x Blocks in the human data.

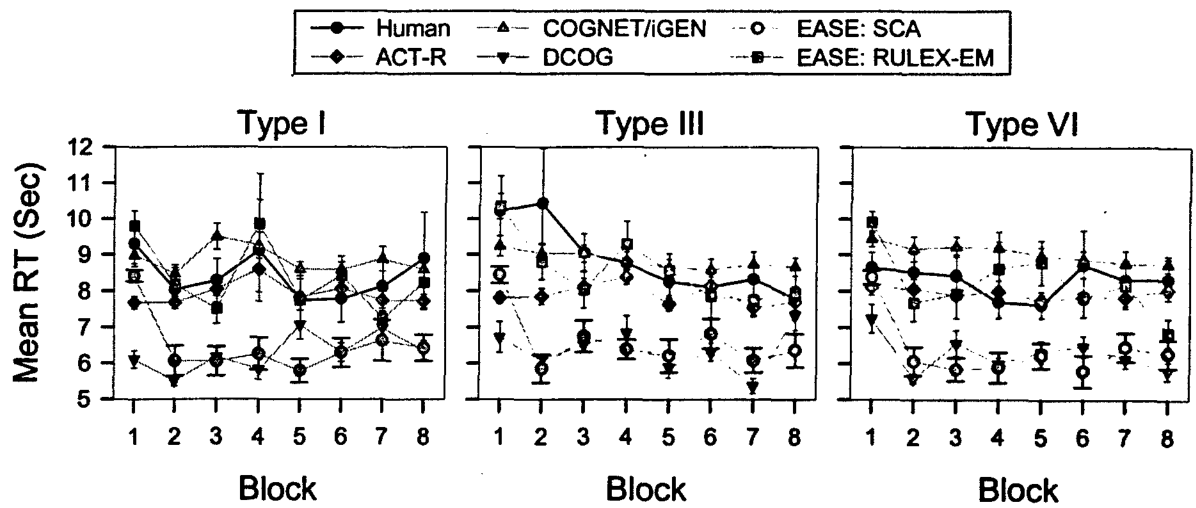


Figure 22. Human and revised model response times to the handoff task as a function of category learning problem type.

Congruent with the human results, the revised versions of the models all showed a significant blocks effect. Unlike the human results, DCOG also exhibited an interaction between problem type and blocks. G^2 values were 14.20 (COGNET/iGEN), 15.24 (EASE RULEX-EM), 21.25 (ACT-R), 114.73 (EASE SCA), and 135.83 (DCOG). Response times by the DCOG and EASE

SCA models were too fast, contributing to their high SSE value. The other models showed reasonably good fits to the observed data.

4.1.2.3 Transfer Task

We begin our analysis of the transfer data by first comparing performance on the "Trained" transfer items that had previously been encountered, with performance on those same items from the last block of training. We contrast this with performance on "Extrapolated" transfer items more extreme than the "Trained" items. Extrapolated stimuli were scored in the same manner as the nearest previously trained item. The "Trained" vs. "Extrapolated" comparison was designed to assess how well strategies generalized from one type of item to another. The "Last Block" vs. "Trained" comparison allowed for an evaluation of how well performance transferred from the learning portion of the experiment to the transfer condition. Results from the observed data, initial model predictions (made prior to having the observed human results), and revised model data are shown in Figure 23 for each Problem Type.

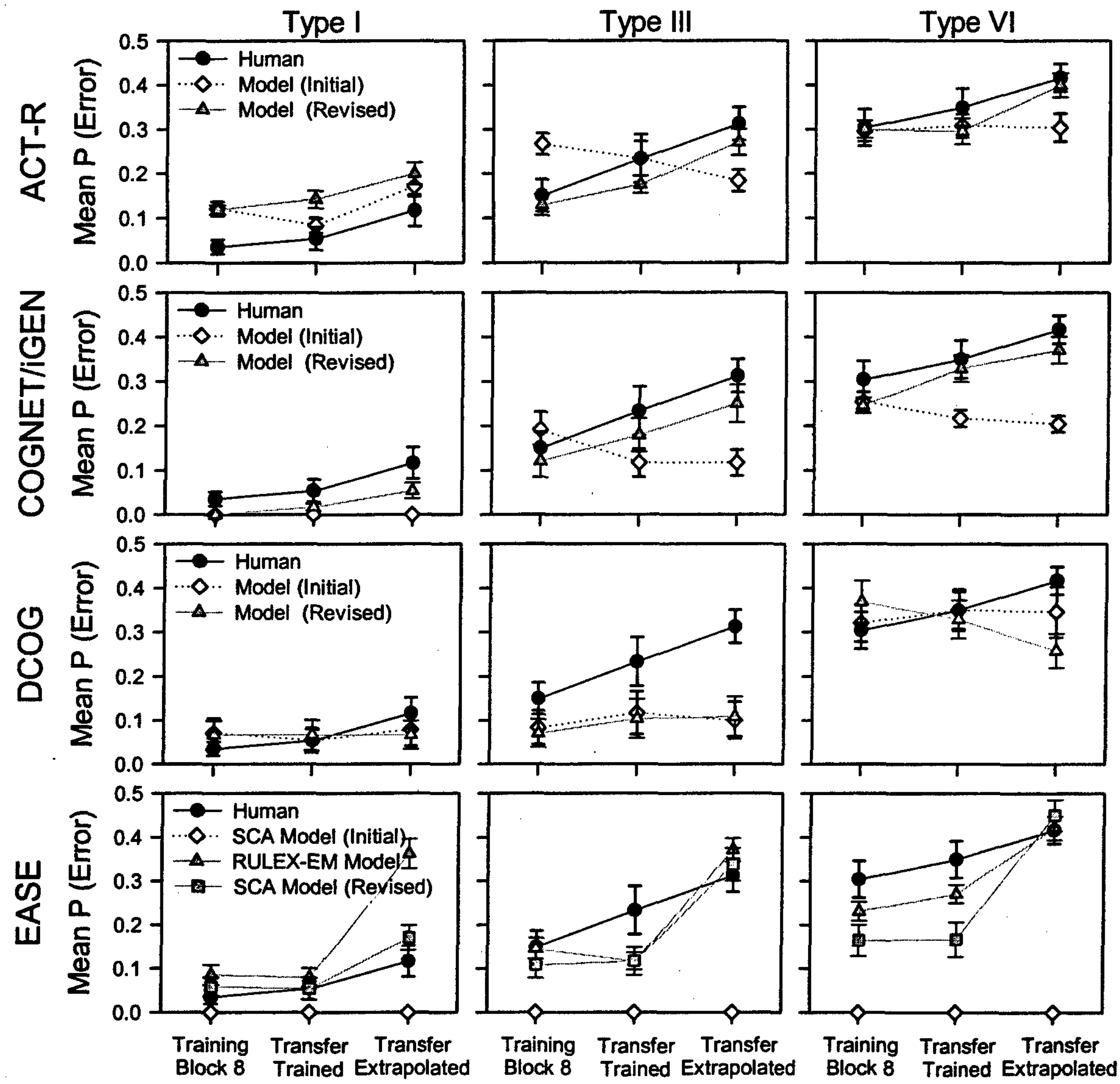


Figure 23. Human data, initial model predictions, and revised model data for block 8 learning data, trained, and extrapolated transfer test stimuli.

An ANOVA on the human data showed there were significant main effects of problem type and items, where the three possible levels of items are Training Block 8, Transfer Trained, and Transfer Extrapolated (See Table 9). The pattern of results shows there was a significantly greater number of errors on the Trained items on the transfer test than on the identical items in Training Block 8. Less surprising was the finding that Extrapolated items were missed more frequently than previously trained items on the transfer test. A Tukey analysis showed that all three types of items differed significantly from each other ($p < .05$).

Table 9: A Comparison of Human Data, Original Model Predictions, and Revised Model Data for Transfer Task Analysis of Trained and Extrapolated Items

	ANOVA Main Effects and Interactions		
	Problem Type	*Items*	*Problem Type by Items*
Human	***Significant****	***Significant****	***Not Significant***+
Original Model Predictions			
ACT-R	✓		✓
COGNET/IGEN	✓	X[a]	
DCOG	✓		✓
EASE SCA	Not Computable[b]	Not Computable[b]	Not Computable[b]
Revised Model Predictions			
ACT-R	✓	✓	✓
COGNET/IGEN	✓	✓	✓
DCOG	✓		
EASE SCA	✓	✓	
EASE RULEX-EM	✓	✓	

* $p < .0001$, + $p > 0.05$

[a] *Incorrect direction of effect*

[b] *Due to a lack of variance (no simulated subjects made any errors, so the variance was 0) F values could not be computed.*

We calculated a G^2 value to determine how well each model predicted these data, without prior knowledge of the observed results: 11.01 for ACT-R, 16.77 for DCOG, 48.96 for COGNET/iGEN, and 420.09 for EASE SCA. The results of ANOVAs performed on the initial model results to look for significant effects of items and problem type are described in Table 9. All models showed the desired problem type effect. However, all the models initially failed to produce the observed items effect. COGNET/iGEN produced a significant items effect that was reversed, with the highest probability of error for the Training Block 8 items instead of the Extrapolated items for Problem Types III and VI. COGNET/iGEN also showed an undesirable, significant interaction of Problem Type x Items, reflecting the floor effect seen in Problem Type I. As shown in Figure 23, DCOG's curves appear flat across item types. Unlike the observed data, EASE SCA showed perfect performance on all three types of items for all Problem Types. F statistics could not be computed for the EASE SCA results due to this lack of variability.

After making their predictions, the modelers were given the observed data from the transfer task and allowed to revise their models. Most of the model fits improved, some considerably, after model revisions, as illustrated by the following G^2 values: 7.99 for ACT-R, 8.53 for COGNET/iGEN, 14.37 for EASE SCA, 16.23 for EASE RULEX-EM, and 21.28 for DCOG.

DCOG's fit was slightly worse than their initial prediction, showing the desired problem type effect, but still not producing the items effect. This time there was an undesirable interaction of Problem Type x Items. As shown in the graphs, there was no items effect for Problem Types I and III, and a reversed items effect for Problem Type VI (See Figure 23). COGNET/iGEN improved its fit, showing the observed items and problem type effects with no interactions. ACT-R improved its fit, this time showing significant effects of items and problem type. However, as is clear from the graph, ACT-R under-performed in Problem Type I. EASE SCA made a huge improvement in their fit to the human data. EASE SCA exhibited a significant effect of problem type, but showed a significant interaction of Problem Type x Items. While showing a significant effect of items, EASE SCA did not show the observed decrease in performance in the transfer condition on previously trained items, relative to performance on block 8. EASE RULEX-EM also showed a similar pattern of results, with the desired effect of problem type and items, but an undesired interaction of Problem Type x Items. As seen in the graphs, EASE RULEX-EM showed the desired performance decrement on old items in the transfer condition in Problem Type VI condition, but not in the easier conditions, and showed poor performance on Problem Type I extrapolated items.

4.1.2.4 Subjective Workload Ratings

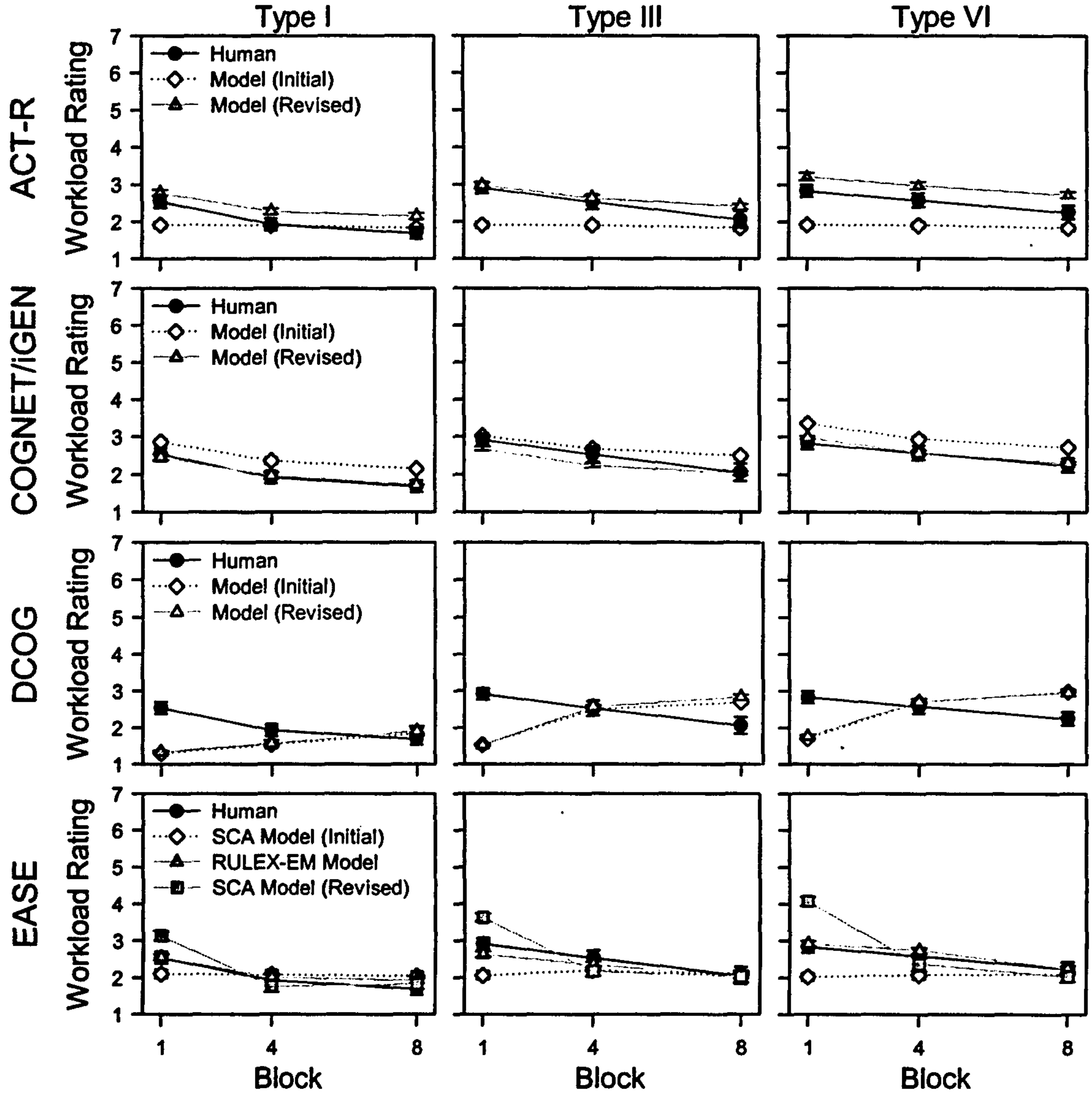


Figure 24. Observed and predicted subjective workload ratings administered after blocks 1, 4, and 8.

Figure 24 illustrates subjective workload ratings taken after blocks 1, 4, and 8 for both the initial and revised model predictions as compared to the human data. Table 10 shows that initial predictions by all models showed significant effects of secondary task workload, contrary to the observed data. Also, unlike the human results, initial predictions by ACT-R and EASE SCA showed no effect of problem type. In addition, EASE SCA failed to predict the block effect. Although ACT-R showed a main effect of block, this was an artifact of producing extremely low between-subjects variability, producing a large F value, even though visual examination of Figure 24 shows virtually no effect of block. The initial predictions of COGNET/iGEN showed

both the problem type and block effects. DCOG showed the desired problem type effect, but had a reverse block effect, predicting a workload rating increasing over blocks, unlike the observed results. Additionally, DCOG was the only model to predict an interaction of Problem Type x Block; an interaction not seen in the human results. SSE values for the initial model predictions are as follows: 1.34 for COGNET/iGEN, 2.13 for EASE SCA, 3.37 for ACT-R, and 5.89 for DCOG.

Table 10: A Comparison of Human Data Results and Model Predictions for Workload Ratings

	ANOVA Main Effects and Interactions				
	Problem Type	*Block*	*Secondary Workload*	*Prob. Type by Block*	*Workload by Block*
Human	***Significant****	***Significant*****	***Not Sig.***$^{+}$	***Not Sig.***$^{+}$	***Significant*****
Original Model Predictions					
ACT-R		✓		✓	
COGNET/IGEN	✓	✓		✓	
DCOG	✓	X[a]			✓
EASE SCA				✓	
Revised Model Predictions					
ACT-R	✓	✓			
COGNET/IGEN	✓	✓	✓	✓	
DCOG	✓	X[a]			✓
EASE SCA	✓	✓			
EASE RULEX-EM	✓	✓			

$^{*}p < .05$, $^{**}p < .0001$, $^{+}p > 0.05$

[a] *Incorrect direction of effect*

The model revisions achieved closer fits in all cases. The final SSE values were: 0.21 for EASE RULEX-EM, 0.33 for COGNET/iGEN, 1.05 for ACT-R, 2.66 for EASE SCA, and 5.83 for DCOG. All models, with the exception of DCOG, showed a block effect in the correct direction. DCOG still predicted an increasing workload difficulty rating. Additionally, all models showed the effect of problem type, with Problem Type I having a lower workload than Problem Type III or VI. All models, except COGNET/iGEN, continued to show an unhuman-like effect of secondary task workload. Additionally, only DCOG showed a secondary task workload by block interaction. However, this was offset by DCOG's incorrect prediction of an interaction between

problem type and blocks. In fact, model revisions resulted in ACT-R, EASE SCA, and EASE RULEX-EM showing unhuman-like Problem Type x Block interactions; something not seen in their initial models.

4.1.3 Summary of Model Fits

These models' ability to postdict and predict (in the case of the transfer task predictions) the primary human results is summarized in Table. The table shows each model's "best shot" at replicating the results, after the opportunity for revisions, except in the case of the transfer task predictions, which include both initial and revised model predictions.

Table 11: Summary of Model Comparison Results

	Predicted Result				
Desired Result Type	*ACT-R*	*COGNET/ IGEN*	*DCOG*	*EASE SCA*	*EASE RULEX-EM*
Primary Task Learning Results					
Problem Type Effect	✓	✓	✓	✓	✓
Block Effect	✓	✓	✓	✓	✓
No Type by Block Int.		✓			
Problem Type III Central vs. Peripheral Item Results					
Item Effect	✓	✓	✓	✓	
Block Effect	✓	✓	✓	✓	✓
No Item by Block Int.				✓	
Primary Task Response Time Results					
Problem Type Effect					
Block Effect	✓	✓	✓	✓	✓
No Type by Block Int.	✓			✓	✓
Secondary Task Penalty Point Results					
No Prob. Type Effect	✓	✓	✓	✓	✓
No Prob. Type Effect	✓	✓		✓	✓
Other					Too few points

	Predicted Result (continued)				
Desired Result Type	*ACT-R*	*COGNET/ IGEN*	*DCOG*	*EASE SCA*	*EASE RULEX-EM*
Secondary Task Response Time Results					
No Prob. Type Effect	✓	✓	✓	✓	✓
Block Effect	✓	✓	✓	✓	✓
No Type by Block Int.	✓	✓		✓	✓
Other			Too slow	Too slow	
***Transfer Task* Prediction**					
Problem Type Effect	✓	✓	✓	X[b]	--[c]
Items Effect		X[a]		X[b]	--[c]
No Type by Items Int.	✓		✓	X[b]	--[c]
***Transfer Task* Postdiction**					
Problem Type Effect	✓	✓	✓	✓	✓
Items Effect	✓	✓		✓	✓
No Type by Items Int.	✓	✓			
Workload Rating					
Problem Type Effect	✓	✓	✓	✓	✓
Block Effect	✓	✓	X[a]	✓	✓
No Secondary Workload Effect		✓			
No Type by Block Int.		✓			
Workload by Block Int.			✓		
Percent Match	74.1%	77.8%	51.9%	63.0%	62.5%

[a] *Incorrect direction of effect*

[b] *Due to a lack of variance (no simulated subjects made any errors, so the variance was 0) F values could not be computed.*

[c] *EASE RULEX-EM did not exist in the set of initial model predictions.*

As can be seen in the table, ACT-R matched 74.1% of the main results. In the primary task, ACT-R showed an interaction of Problem Type x Block, unlike the human results. Also, it did not show an effect of problem type in response times in the primary task (i.e., it did not show an increase in reaction times with increased problem type complexity, as seen in the observed data). Like most other models, ACT-R showed a number of problems with predicting workload rating results: predicting a secondary workload effect, and interaction of Problem Type x Block, but no interaction of Secondary Task Workload x Block.

Overall, COGNET/iGEN showed the best fit to the data, matching 77.8% of the primary results and having the best G^2 or SSE on 4 of 8 performance measures. Like ACT-R, COGNET/iGEN did not show an effect of problem type in primary task response time results. Additionally, it predicted an unobserved interaction of Problem Type x Block for the primary task response time measure and also incorrectly predicted an unobserved interaction of Item x Block as part of the detailed analysis of Problem Type III.

DCOG matched 51.9% of the main findings; with most of its incorrect predictions due to predicting unobserved interactions between effects, including 1) Problem Type x Block in primary task accuracy, 2) Problem Type x Block in primary task reaction time, 3) Problem Type x Items in the transfer task, 4) Problem Type x Item in the detailed analysis of Problem Type III, 5) Problem Type x Block in secondary task response time, and 6) Problem Type x Block in subjective workload ratings. Finally, DCOG predicted an unobserved secondary workload effect on subjective workload ratings, and did not predict an effect of problem type on primary task response time results.

EASE SCA matched the observed data on 63.0% of the primary indicators, producing a reasonable qualitative fit to much of the data. Like the three other models, EASE SCA predicted an unobserved interaction of Problem Type x Block on the category learning accuracy measure. Unlike any other model, EASE SCA correctly replicated all three findings in the detailed analysis of Problem Type III. EASE SCA did not predict the effect of problem type on the primary task reaction time measure seen in the observed data. EASE SCA consistently under-predicted secondary task reaction times, and predicted an interaction of Problem Type x Items in the transfer task, unseen in the human data. EASE SCA also mismatched the workload ratings findings in a manner consistent with ACT-R and EASE RULEX-EM: incorrectly predicting a

secondary task workload effect and an interaction of Problem Type x Block, but not predicting a secondary task workload interaction with blocks.

EASE RULEX-EM matched 62.5% of the principal results[5]. EASE RULEX-EM's pattern of results was typical in many respects. It showed the same pattern of workload rating results as ACT-R and EASE-SCA. Like ACT-R, DCOG, and EASE-SCA it predicted an unobserved interaction of Problem Type x Block for the primary task accuracy measure and did not predict an observed problem type effect in primary task reaction times. EASE RULEX-EM was the only model not to predict a significant difference between central and peripheral items in Problem Type III. Like DCOG and EASE SCA, it predicted an unobserved interaction of Problem Type x Item's in the transfer task.

There were a number of surprising findings in the modeling predictions. First, no models predicted differences in primary task response times across problem types, unlike humans, who produced longer response times for the harder problem types (III & VI) than for the easiest problem type (I). Second, no model initially predicted the decrease in performance accuracy on trained items when presented in the transfer condition, and the COGNET/iGEN model actually predicted the *opposite* of the observed effect (predicted performance improvement, when in fact humans showed a performance decrement in the transfer condition). However, model revisions for ACT-R and COGNET/iGEN models were able to reproduce the general pattern of results. Third, no model initially predicted worse performance on the new extrapolated items, relative to previously observed items. Lastly, we were surprised at the initial deviations from the category learning curves by many of the models. ACT-R initially learned too slowly in Problem Type I, while EASE SCA learned much too quickly in all problem types. Additionally, DCOG produced a nonmonotonic learning curve in Problem Type VI.

4.2 Other Factors in Model Comparison

The AMBR models showed varying qualitative and quantitative fits to an assortment of data subsets (see Table and the goodness-of-fit measures earlier in this chapter), with the revised COGNET/iGEN model postdicting the highest overall percentage of the qualitative effects and the DCOG model postdicting the lowest overall percentage. This begs the question, "Is the COGNET/iGEN model a better model than the DCOG model?" Or the EASE models? Or the

[5] Transfer task prediction results were not included in the evaluation of RULEX-EM because the model was developed after the release of the transfer data to the modelers.

ACT-R model? As we discuss in this section, the quality of a model is very much in the eye of the beholder. The quality/acceptability/appropriateness of a model, and any effort to rank order it relative to other models developed with other architectures, depends very much on what one values in a model and in an architecture. There are several other factors to consider in evaluating and comparing models.

Thus far, the comparison of the AMBR models has focused entirely on the quality of their fits to the experimental data. It would be easy for the reader to get the mistaken impression that it is our position that a comparison of these models should be based entirely on goodness of fit, and that goodness of fit to empirical data is the most important dimension on which to compare models. That is not the case. In a recent series of articles, Roberts and Pashler (2000, 2002) and Rodgers and Rowe (2002a, 2002b) debated the role of goodness of fit in theory testing. Roberts and Pashler (2000) started the discussion with an attack on goodness of fit as the metric for assessing the quality of psychological theories. Rodgers and Rowe came to the defense of goodness of fit, and by the end of the interaction, all parties seem to have agreed that goodness of fit measures serve as a good starting point (but not ending point) in the evaluation process. We strongly agree that quantitative and qualitative measures of goodness-of-fit are good starting points in evaluating models. We're also quick to point out that there are several other important factors one might consider when comparing models of human behavior. These include the degrees of freedom available in implementing the model, how much of the model was reused from previously implemented models, the interpretability of the model's behavior during run time, and the generalizability of the model. Below we discuss why each of these is a dimension of interest and how the AMBR models compare on each of them.

4.2.1 Degrees of Freedom

The degrees of freedom available during model implementation are important to consider because they provide a context in which to interpret the impressiveness of a particular fit. There is a positive correlation between degrees of freedom and expectations for fit statistics. As the degrees of freedom increase, so should goodness-of-fit, because the modeler has a great deal more flexibility in the implementation of the model.

Researchers involved in mathematical modeling of psychological phenomena often emphasize the importance of considering degrees of freedom during model evaluation (e.g., Myung, 2000; Pitt, Myung, & Zhang, 2002). They generally refer to this issue as one of

complexity, and numerous approaches are available for quantifying the complexity, or degrees of freedom, available in a closed-form mathematical model (e.g., Bozdogan, 2000; Busemeyer & Wang, 2000; Grünwald, 2000; Wasserman, 2000).

The AMBR models are not closed-form mathematical models, but this doesn't mean that complexity is not an issue. It just means we need to consider alternative approaches to identifying and quantifying degrees of freedom. In the behavior representation and cognitive modeling communities a distinction is often made between the architecture (relatively fixed structure) used to develop models and the knowledge that we represent with those architectures in order model behavior in a specific context or domain. This creates a useful classification scheme for degrees of freedom in computational process models like those developed for AMBR: architecture degrees of freedom and knowledge degrees of freedom.

4.2.2 Architecture

Creating entirely new architectural capabilities, like spreading activation where none existed before, or an instance-based learning mechanism where none existed before, is a powerful way to achieve additional degrees of freedom in the implementation of a model. There are so many decisions made in the implementation details of new architectural capabilities that it might be fair to consider it to be the case that they involve multiple additional degrees of freedom. Among the AMBR models, the ACT-R architecture added no new capabilities (all necessary modeling capabilities were already in place), COGNET/iGEN and EASE both added new learning capabilities, and the entire DCOG architecture was under development. Lest the reader get the impression that the previous sentence was a rank ordering of the degrees of freedom associated with the use of each of these architectures, we should note that not all architectural additions are created equal, in terms of how much freedom they provide the modeler. Things like strict adherence to specific theoretical constraints and/or code reuse from other models can have a significant impact on the degrees of freedom associated with the implementation of any particular architectural feature. An example of this is that the EASE SCA model borrowed an existing learning mechanism from SCA-Soar, thereby tightly constraining the addition of their new architectural learning capability. By contrast, the COGNET/iGEN learning mechanism was constructed from scratch and was not strictly constrained by specific theoretical commitments, which provides quite a lot of freedom in its implementation.

An alternative to creating architectural features is removing them, which serves as a second type of architectural degree of freedom. For example, Soar's powerful learning mechanism (called "chunking") was deactivated in the TacAir-Soar model (Jones et al., 1999) because that was considered to be a model of expert performance and was to be used in situations where new learning was considered unimportant and perhaps even undesirable. In ACT-R modeling, it is customary to deactivate components of the architecture that are not central to the psychological focus of a particular model. Irrespective of the architecture, any time the modeler is making a choice about activating or deactivating an architectural component, it is a degree of freedom in the model's implementation. A caveat is when it is conclusively demonstrated that the performance of the model is entirely insensitive to the presence of a particular architectural component, in which case that component arguably is not a degree of freedom.

Numerical parameters are a third type of architectural degree of freedom. These are things like retrieval threshold (ACT-R and EASE) and patience (DCOG). Interestingly, the COGNET/iGEN team reports that COGNET has no numerical parameters in its baseline architecture (Chapter 6, Gluck and Pew, in press). It is important to point out, however, that they make up for this through the use of micromodels that are tailored to the demands of each specific modeling context in which COGNET/iGEN is used. Additionally, COGNET/iGEN did add three numerical parameters in their implementation of a learning capability. If these parameters are used in future COGNET/iGEN models, perhaps they will come to be seen as architectural parameters.

Among the three teams who do report having architectural parameters available to them in the AMBR models, ACT-R used all default values for the multi-tasking model in Experiment 1, but changed three architectural parameters (retrieval threshold, value of the goal, and goal activation) for the category learning model in Experiment 2. DCOG used one free parameter (time-factor) for the multi-tasking model in Experiment 1 and seven additional free parameters for the category learning model – to create individual differences among operator representations. EASE used all default architectural parameters for the multi-tasking model in Experiment 1, but EASE-SCA and EASE-RULEX-EM each were allowed one free parameter for the category learning model in Experiment 2.

4.2.3 Knowledge

Architecture-based computational process modeling involves adding knowledge to the architecture to get behavior in specific contexts. The architecture is supposed to constrain the allowable structures in the knowledge (e.g., production rules, chunks, operators, frames) but does not necessarily constrain the content of those structures. It is in the knowledge where task strategies, domain expertise, and general knowledge are implemented. There are (potentially) both symbolic and numeric degrees of freedom in knowledge representation, such as activation values for declarative chunks or utilities for production rules, to draw on a couple of examples from ACT-R. Baker, Corbett, and Koedinger (2003) propose some guidelines for quantifying knowledge degrees of freedom (i.e., parameters) for comparing different models developed in ACT-R. The basic approach is that they count every production rule and chunk that influences the behavior of the model and they also count every numerical parameter that is not fixed to some default or other a priori value. As one would expect, this results in large numbers of parameters, even for relatively simple domains. No doubt, any similar exercise undertaken with COGNET/iGEN, DCOG, or EASE would also result in large numbers of parameters for models developed with those architectures. This might be a fun exercise, but comparing the results across the AMBR models would be misguided. Baker et al. note that differences in representational granularity make it inappropriate to compare models written in ACT-R 4.0 to those written in ACT-R 5.0. In other words, because ACT-R 4.0 represents cognition at a coarser granularity while ACT-R 5.0 represents cognition at a more atomic level of representation, counting free parameters in the manner suggested by Baker et al. necessarily results in a higher number of free parameters in ACT-R 5.0 models than ACT-R 4.0 models. We have the same problem in comparing across the AMBR architectures. The different architectures modeled performance and learning at different granularities, and so a quantitative comparison of knowledge degrees of freedom would be inappropriate and misleading.

4.2.4 Model Reuse

Code reuse is highly desirable in software engineering because it increases cost effectiveness and standardization. Model reuse is highly desirable in human behavior representation for the same reasons, and also because it can teach us something about the generalizability of the representations used in other models (more on generalizability in a moment). However, model reuse is very difficult to accomplish, and is almost never done in any large-scale way. This is

because human performance and learning occur at the intersection of knowledge and environment, and as the context and task domains vary from one model to the next, the knowledge in the model must also vary. The knowledge required for Task B must be almost identical to the knowledge required for Task A in order to have any chance of successfully porting the model for Task A over to Task B. The pattern of model reuse (or lack thereof) was fairly predictable in the AMBR Model Comparison, with a couple of exceptions. *None* of the models reused existing code for the multi-tasking model in Experiment 1. *All* of the models reused their code from the Color/Aided condition in Experiment 1 when implementing their category learning models for Experiment 2, except for the DCOG team, who reimplemented their entire architecture in Java (from Lisp) during the transition from Experiment 1 to 2. The EASE SCA model also reused the SCA model (i.e., the production code) developed by Miller and Laird (1996). The developers made only the minimum number of changes to the model for the current version of EASE (SCA was originally developed in a much earlier version of Soar) and they confirmed that the learning results generated by the new model were exactly the same as those produced by Miller and Laird (1996).

4.2.4.1 Interpretability

Model interpretability is a significant issue for human behavior representation models, in the sense that it typically is difficult to know why the model is doing what it is doing, and sometimes is even difficult to know *what* it is doing at a particular time. This is an issue of run-time interpretability, and it plagues all human behavior representation architectures. Only the EASE team took steps to address this issue during the AMBR Model Comparison. They did so by adding a color-coded legend to the task display that marked which of several possible cognitive activities was taking place at any moment in time. It served not only as a helpful debugging tool for the model developers, but also as a helpful learning tool for those trying to become familiar with the model's implementation.

4.2.4.2 Generalizability

This final factor to consider in comparing computational models is really quite simple. We would like for it to be the case that a model that is developed for, or fit to, one set of data will generalize to another set of data. We would like for it to be the case that model predictions (or postdictions) extrapolate with some predictive accuracy to contexts/situations/stimuli beyond those for which the model was specifically developed. Our modest attempt at pushing the models

in this direction during Experiment 2 revealed that the field has quite a lot of room for improvement in this area. None of the models accurately predicted even the *direction* of the results in the transfer condition.

4.3 Model Architectural Comparisons: The Seven Common Questions

This section examines the similarities and differences found across the four model architectures employed in the AMBR project with the goal of illustrating the architectural implications for modeling multi-tasking and category learning phenomena. We base this discussion around a set of seven questions given to each modeling team – the answers to which are presented at the end of each of the modeling chapters (Chapters 4-7, Gluck and Pew, in press). Before comparing responses to each of the seven questions, it is useful to examine the historical origins and theoretical assumptions of these model architectures, as it provides insight into their fundamental capabilities and their architectural strengths and weaknesses.

EASE is the latest in a line of hybrid models, with its roots in the Soar architecture (Newell, 1990), but also borrowing and integrating elements from EPIC (Kieras & Meyer, 1997, 2000) and ACT-R (Anderson & Lebiere, 1998) in order to augment less well-developed portions of the Soar architecture[6]. Work on the Soar architecture has historically focused on developing intelligently behaving systems, with less emphasis on modeling detailed psychological phenomena. That is not to say, however, that Soar and Soar hybrids such as EASE have not been used to model detailed psychological results – they have – only that this is more the exception than the rule in the Soar community and the historical bias has been on general intelligence, rather than specifically human intelligence, with an emphasis on demonstrating mechanisms and functions sufficient for general intelligence.

ACT-R, on the other hand, has its roots in psychological theories of memory, learning, and problem solving (Anderson, 1983, 1990, 1993). ACT-R places an emphasis on modeling task accuracy and reaction time, incorporating both symbolic and subsymbolic mechanisms. Like Soar, ACT-R has a long research tradition and has been used to explain a range of cognitive phenomena[7].

Unlike EASE or ACT-R, COGNET/iGEN has not grown out of any desire to develop a general approach to psychological theory or cognition, but was developed as a framework for the

[6] EASE extends its predecessor, EPIC-Soar, by adding elements of ACT-R to the EPIC/Soar hybrid model.
[7] See the ACT-R website for a fairly comprehensive list of phenomena and task domains (http://act.psy.cmu.edu)

development of human behavior representations in practical, real-world applications such as intelligent interfaces, and training and decision-support systems. COGNET/iGEN was designed to provide a great deal of flexibility with which to create models, and is intentionally theory-neutral with respect to many of the underlying processes.

DCOG does not have an extensive history. It was actually under development during the AMBR Comparison. Unlike the other three architectures, DCOG does not advocate an information processing viewpoint, but describes itself as a distributed, state-change system where "mind states themselves give rise to information, based on energetic stimulation from other local mind regions and the external environment" (Chapter 6, Gluck and Pew, in press). DCOG is also unique in its implementation as a distributed software agent architecture. As a new modeling architecture, DCOG is less comprehensive than the other model architectures, and it has not been evaluated against behavioral or cognitive phenomena outside the AMBR project.

4.3.1 The Seven Questions

1. How is cognition represented in your system?

4.3.1.1 Perception

The EASE model has the most highly developed perceptual system of any model in the AMBR project. Based on the EPIC model, EASE's visual processing system represents retinal processing limitations as well as eye scan patterns, which are based both on top-down and bottom-up processing. Limitations on featural perception are modeled through the use of retinal zones; with certain classes of features such as object direction requiring foveal processing, while other events such as stimulus onsets processed in all retinal zones. Additional limits are due to perceptual memory decay mechanisms. Eye scan patterns are based on priority values associated with perceptual events. Both perceptually-based and knowledge-based priorities are represented, with precedence given to knowledge-based priorities. Explicit strategies for scanning message history lists were also developed.

The COGNET/iGEN system developed visual scanning mechanisms based on cognitive task analysis (CTA), leading to the development of simple scanning mechanisms, which assumed changes in a display pane can be processed directly and in their entirety. Additionally, CTA suggested no perceptual memory mechanisms were required, and none were implemented. Different scanning strategies were developed for the text and color display conditions. The color display scanning strategy contained only a single goal of detecting color changes in the radar

display. The text display condition incorporated a more complex strategy involving scanning the radar display for red stimuli, followed by checking the text panes in a fixed sequence for information different from that found in memory.

Perceptual processing in ACT-R was implemented as a set of production rules and subgoals that systematically scanned the display panes and added display information to memory. In order for the model to respond to event onsets, a new visual onset detection mechanism was implemented with a number of processing limitations. First, onsets can only be detected during a limited time window, and if the system is busy during that time, the onset will not be detected. Second, only a single onset event will be detected during a production cycle – subsequent onsets will be ignored.

In the DCOG model, an agent is assigned the task of monitoring and processing the display and making perceptual information available to other agents in the system. The Radar agent stores perceptual features, such as color and aircraft name into an iconic memory. Additionally, higher knowledge-level, task relevant events such as the aircraft is entering or leaving the airspace are encoded and stored in a global memory. A visual scanning strategy was implemented that scans the four boundary regions in the radar display followed by the text message history panels. Scanning sequences repeat every five seconds. A scanning sequence can be interrupted and attention paid to a specific display region when the number of aircraft soon to require attention exceeds the worry-factor strategy variable.

4.3.1.2 Knowledge Representation and Cognitive Processing

COGNET/iGEN distinguishes among and represents five different types of knowledge or expertise: declarative, procedural, action, perceptual, and metacognitive expertise. Declarative and metacognitive memory elements are represented in separate blackboard systems. Procedural, action, and perceptual expertise are represented using GOMS-like goal hierarchies and a specialized task description language. The cognitive processor executes a single cognitive task at a time, but there can be multiple active tasks in various states of completion. Tasks are activated when its conditions are met by elements in memory, and task completions are a means of accomplishing a goal. Fast task switching simulates multi-tasking capabilities.

In ACT-R, declarative knowledge is represented as structured memory elements, or chunks, while procedural knowledge is represented as production rules. Cognitive processing is a function of activity at both the symbolic and subsymbolic levels. At the symbolic level, ACT-R

is a serial-firing production rule system where all productions whose conditions match elements in memory are instantiated, but only a single production is selected using ACT-R's conflict resolution mechanisms and fired. Subsymbolic mechanisms determine the speed and success of memory access, and also participate in conflict resolution mechanisms. Cognition is goal-driven and a goal stack is used to track goals in the AMBR models, which were implemented in ACT-R 4.0. More recent releases of ACT-R do not include a goal stack. The goal stack was used in the AMBR models for historical reasons and really played a minor role. In particular, the goal stack was NOT used to remember where to restart processing after handling an interruption.

Like ACT-R, EASE represents declarative knowledge as structured memory elements, and procedural knowledge in production rules. While EASE's cognitive processing system is also a production rule system, in contrast to ACT-R, it is a parallel firing production system where every rule that matches is fired. Rules are used to propose or register preferences for operators – of which, only a single operator is selected using conflict resolution mechanisms and fired. EASE integrates ACT-R's subsymbolic memory mechanisms for improved modeling of memory effects. Like ACT-R, control is organized around a goal hierarchy. Unlike ACT-R, where goals arise from productions, in EASE, goals or subgoals are created when an operator cannot be selected, resulting in an impasse.

DCOG was designed as a framework for developing software agents to model human performance. The framework is based on four principles: 1) distributed knowledge and control, 2) emergent forms of knowledge, 3) communication through broadcast signaling, and 4) cognitive strategies form the basis for complex behaviors. DCOG views cognitive processing as a state-change system, with software agents executing parallel computational threads of activity. Knowledge emerges as a pattern of activations over distributed regions.

4.3.1.3 Memory

Memory in the DCOG-2 model is based around an associated memory system where feature-based-knowledge and symbol-based knowledge, such as stimulus exemplars and hypotheses, are represented as nodes in the system. The co-activation of nodes forms and strengthens associative links among the nodes, providing pathways for spreading activation across nodes. Procedural or functional knowledge is stored as procedures executed by a software agent.

In COGNET/iGEN, memory elements are represented within a blackboard, with a number of separable and distinct areas for different classes of information (e.g., perceptual, domain

knowledge). COGNET/iGEN also postulates a metacognitive memory representing the state of the cognitive, perceptual, and motor systems. A number of extensions were developed to the COGNET/iGEN memory systems for the AMBR Comparison, specifically to support the learning mechanism and were only used within the learning mechanism. In those extensions, separate memory systems were developed for short-term and long-term memory and memory constraints were implemented based on the principles of decay, rehearsal, and proactive interference. Memory elements are maintained in short-term memory through rehearsal, and each rehearsal provides an opportunity to transfer the element to long-term memory, which does not decay. Retrieval from short-term memory is based on the complexity of the memory element and the amount of rehearsal afforded it.

ACT-R contains three separate memory structures. Declarative memory consists of chunks, or memory elements, with activation levels and weighted associations to other chunks. Procedural memory is made up of production rules. Together, these make up long-term memory. Finally, a last-in-first-out goal stack is used to track goals and guide behavior.[8] The goal stack and the most active declarative memory elements make up working memory. Memory limitations are based on sophisticated subsymbolic quantities, which represent chunk activation and production utilities.

EASE's memory mechanisms are similar to those found in ACT-R. Like ACT-R, procedural memory consists of rules or operators representing task behaviors. Declarative memory, or working memory, contains memory elements obtained directly from sensory subsystems or through the firing of production rules. ACT-R's subsymbolic chunk activation components are incorporated in order to provide limits to working memory. A goal stack is also used to track goals and focus problem solving.

4.3.1.4 Learning

The AMBR project required the addition of a learning mechanism to COGNET/iGEN. Although specific to the category learning paradigm, its mechanisms for learning the goals and actions to be undertaken were designed to be general and extensible to other forms of learning. A separate category-learning module was developed, with access to short and long-term memory structures,

[8] The goal stack is no longer an architectural feature in ACT-R 5.0, but that version was not available when the AMBR Model Comparison started and Lebiere chose to stick with 4.0 throughout the project. Lebiere reports that the goal stack was not used as a significant memory structure in the AMBR models (personal communication, July 27, 2004).

which learned category representations using a rule-based hypothesis testing approach. The COGNET/iGEN team added a decay mechanism as a memory moderator.

Learning in DCOG is based on building activations and associations among nodes in its associative memory. DCOG employed four distinct category learning strategies, emphasizing different learning styles observed in individuals. In every learning style, associations are built up between units representing response categories and other knowledge structures such as primitive features, exemplars, or category hypotheses.

Learning in ACT-R is a fundamental process, with most components of the model able to be learned, including rules, memory elements, and subsymbolic values. Learning in the AMBR model is based on the learned utility of production rules as well as the activation and associative strengths of declarative memory instances based on their usage history. Limits to learning capabilities are based on subsymbolic memory decay and utility computation mechanisms, as well as limits on the creation of new memory structures as a function of processing.

EASE incorporates ACT-R's declarative memory learning mechanisms, modulating the availability of declarative memory elements based on the recency and frequency of use. In addition, EASE inherited Soar's learning mechanism, in which the results of a subgoal search can be cached as part of a production rule, eliminating the need to generate a subgoal when a similar situation is encountered.

4.3.1.5 Action

In ACT-R, motor actions are presented by production rules, whose latencies are based on the time required to select and apply the production. The application or execution of a production generally has a default latency of 50 msec. However, certain classes of productions related to motor actions, perceptual encodings, and feedback productions were assigned longer latencies. Latency times were not fixed, but instead drawn from a uniform distribution with a range of +/- 25% around the mean.

In COGNET/iGEN, action procedure latencies are estimated through the use of micromodels, the results of which are used to delay the processing thread the associated amount of time. Different latencies were assigned to different types of actions, with latencies a function of the task load levels and display complexities. Errors were introduced into motor response mechanisms with the possibility of pressing the wrong button.

Actions in EASE were represented by task operators, which made use of motor processing mechanisms incorporated from EPIC. Constraints are placed on these operators, such as requiring the hand and eyes to work together to achieve a behavior. Response latencies are a function of stochastically varied motor response parameters as well as the duration of each production cycle, which was varied uniformly.

The DCOG model does not yet have a principled method for including latency measures and did not predict response time measures for either experiment 1 or 2.

2. What is your modeling methodology?

A fundamental principle for the ACT-R and EASE modeling teams was the importance of working with their respective cognitive architectures in order to develop the most natural and effective model of the task. Lebiere (Chapter 4, Gluck and Pew, in press) states their modeling methodology is "based on emphasizing the power and constraints of the ACT-R architecture", while Chong and Wray (Chapter 7, Gluck and Pew, in press) quote Newell and his imperative to "listen to the architecture". In fact, Lebiere goes on to say that they did not analyze the empirical data and protocols to ascertain the strategies used by human participants, but instead "asked ourselves which ACT-R model would best solve the task given the architectural constraints." This is in contrast to the approach taken by COGNET/iGEN and DCOG, both of which emphasized the important role that detailed analysis of human data played in guiding the implementation of their models.

While the ACT-R and EASE modeling teams emphasize the importance of architectural constraints, the COGNET/iGEN team emphasizes the flexibility of the COGNET/iGEN system, enabling them to develop models at the level of granularity best suited for the task. A guiding assumption was to develop models at the most coarse level of granularity required to achieve the modeling goals. The COGNET model was developed from the top down using the iGEN graphical development environment.

The EASE modeling team emphasizes the careful elaboration of the existing architecture only when it is unable to account for behavioral phenomena. As suggested by the name (Elements of ACT-R, Soar, and EPIC), architectural extension is often performed by integrating elements of other previously validated architectures, inheriting their power and constraints as well as their validity.

3. What role does parameter tuning play?

The role of parameter tuning by each of the modeling teams follows rather consistently from their individual modeling methodologies. Both the ACT-R and the EASE modeling teams tried to work within the constraints imposed by their respective architectures – including the reuse of architectural components and default parameter values. On the other hand, the COGNET/iGEN modeling architecture and tool suite is explicitly designed to be unconstrained with respect to model development – the modeler is welcome to implement whatever is necessary to fit the data, at whatever level of granularity seems appropriate. With DCOG's emphasis on individual differences, parameterization was largely used as a mechanism for developing different model instances representing different populations of individuals.

The presence of preexisting architectural components and their associated parameters enabled both ACT-R and EASE to reuse parameter values. ACT-R used default parameter values in all cases where default values existed, and coarsely set other parameters such as the memory retrieval threshold, perceptual and motor action times, stimulus similarity values, and workload scalers.

EASE reused ACT-R's memory activation and decay mechanisms and carried over the default parameter values. Additional parameters such as aircraft color priorities were coarsely estimated, while others such as the number of rehearsals in RULEX-EM, number of extra features in SCA, and workload scaling factors were fit to the empirical data. COGNET/iGEN also adapted other architectural components – reusing the HOS memory moderation model and carried over two of the four parameter values from prior HOS models.

COGNET/iGEN made more extensive use of parameters as part of micromodels within the architecture. These micromodels contained parameters that reflected perceptual, cognitive, and motor action times, practice effects, confusion factors for degrading transfer task performance, and workload scaling factors.

As the only model not based on a preexisting architecture, the DCOG modelers did not draw on any prior modeling components or their parameters in the development of DCOG. As noted earlier, DCOG did use free parameters to represent individual differences.

Both Lebiere (Chapter 4, Gluck and Pew, in press) and Chong and Wray (Chapter 7, Gluck and Pew, in press) acknowledge that the knowledge structures constructed as part of the model can add additional degrees of freedom to the model. Chong and Wray (Chapter 7, pg 73, Gluck

and Pew, in press) state this most strongly, saying, "The primary 'parameter' in the EASE models is knowledge, in that redesigning and reformulating knowledge can often lead to the greatest differences in performance measures." This is evident by the fact that a number of model revisions in Experiment 2 involved changes to knowledge structures and processes. For example, a rule learning mechanism was added to the ACT-R model in order to account for fast learning in the Problem Type I condition. This point is also illustrated by the construction of two EASE models, EASE SCA and EASE RULEX-EM, as part of the model revision process. Both models were developed within the constraints of the EASE architecture. Although each model used different knowledge structures, both ultimately produced very similar overall fits to the data.

Lebiere (Chapter 4, Gluck and Pew, in press) was unique in providing a detailed analysis of the influence of the three parameters involved in memory retrieval processes on category learning accuracy in category Problem Type I. Lebiere systematically evaluated a range of parameter values, revealing the range of data that can, and cannot, be accounted for by the model.[9]

The lack of quantitative parameter optimizing among these AMBR models is unusual for the development of models of multi-tasking and category learning. The majority of category learning models found in the literature are highly tuned to the data. Models of category learning such as RULEX (Nosofsky, Palmeri, & McKinley 1994b), ALCOVE (Kruschke 1992), and SUSTAIN (Gureckis & Love, 2003; Love & Medin, 1998) all used parameter estimation techniques to precisely determine the best fitting parameter values. The AMBR modelers, for the most part, did not highly tune their parameter values, emphasizing a desire to develop a mechanistic understanding of the phenomena, rather than simply fitting a model to the data. The exception to this is the COGNET/iGEN approach that eschews architectural constraints in favor of implementation flexibility through the use of micromodels.

4. What is your account of individual differences?

It is generally agreed in the cognitive modeling and behavior representation communities that there are two ways to represent individual differences: as knowledge differences and as architectural differences. In order to represent knowledge differences observed in the human

[9] These parameter analyses were performed on the original model, prior to adding production rules for single dimension rules. It is interesting to note that no parameter values shown were capable of producing the fast learning shown for human participants in the Type I problem. However, the addition of the single dimension production rules (a knowledge-level change) produced good fits using default architectural parameter values.

data, several of the modeling teams developed a variety of different strategies, with individual models representing distinct populations of individuals. DCOG made extensive use of strategy differences, employing four distinct learning strategies and mechanisms for shifting between them. EASE SCA developed three different strategy variations for learning which stimulus features to ignore during the category learning experiment. In contrast, EASE RULEX-EM was designed as a normative model and did not attempt to account for individual differences. ACT-R and COGNET/iGEN developed models employing a single strategy.

The second source of individual differences variation employed by a number of the AMBR modelers was the explicit manipulation of architectural parameter values. EASE SCA altered the number of features attended by an individual from zero to three, which when combined with the three strategy variations resulted in a total of twelve different individual models. DCOG developed two strategy-mediating variables (worry-factor and process-two) for DCOG-1 and three personality variables (preference, patience and tolerance) for DCOG-2. They used the three personality variables, which mediated category learning strategy shifts, to produce twelve different category learning profile strategies.

A final source of variability employed in all the models was stochastic noise built into various architectural components. This is a form of the architectural parameter approach to modeling individual differences. ACT-R has global noise parameters that influence chunk activations and production rule utilities at model runtime. Despite only using a single knowledge-level strategy, the ACT-R modeling team took the view that each run of the model was equivalent to a separate human participant run. EASE employs stochastic elements in both perceptual and motor elements, as well as noise in memory activation levels – a property inherited from ACT-R. Both EASE SCA and EASE RULEX-EM employ stochastic mechanisms for feature selection in category learning. COGNET/iGEN also had stochasticity built into its feature selection mechanisms. Additionally, COGNET/iGEN made use of randomness as part of micromodels, including generating motor response errors, and incorporating confusions in transfer task judgments. There are no architectural noise parameters in DCOG.

5. What is your account of cognitive workload?

All of the architectures in the AMBR Model Comparison had to design new workload prediction mechanisms in order to account for the subjective workload ratings in Experiments 1 and 2. None of the architectures had been used to account for workload ratings prior to this, although

COGNET/iGEN had previously developed a representation of metacognition. The definitions and implementations of subjective workload are surprisingly different across the three models[10].

The ACT-R model initially represented workload as a scaled ratio of the time spent on critical tasks (process and scan-text goals) to the total time on task. While this representation was sufficient for experiment 1, it performed poorly in experiment 2, leading to the addition of a success-based measure of effort in which the number of errors were weighted and added to the critical time on task.

EASE defined workload as the realization that an activity or event occurred which indicates some work will need to be performed. Workload is not the amount of time or effort spent performing critical tasks, but rather the perception that there is work to be done. Different kinds of "work" were assigned load values representing its relative importance or urgency. Workload was then implemented as the scaled sum of the total realized load divided by the scenario duration.

DCOG used yet another representation for workload. Their approach was to identify eleven factors that appeared relevant to workload estimation which when present added weighted contributions to the associated NASA TLX scale. These factors included actions such as altitude requests processed, perceived task complexity such as the average number of aircraft on the screen, and performance measures including the number of altitude request errors. The DCOG model did not produce workload ratings for Experiment 1, and although workload estimates were generated for Experiment 2, the modelers suggest that the current DCOG architecture does not yet provide satisfactory representations of metacognitive state.

In contrast to the others, the COGNET/iGEN model had previously developed metacognitive capabilities representing underlying state information through a metacognitive memory system – making it unique among the AMBR models. Also, unlike any other model, COGNET/iGEN produced workload assessments across all six workload dimensions, as defined by the TLX workload scales. In COGNET/iGEN, prediction of subjective workload is a complex computation, taking into account factors such as the weighted time spent performing actions, the number of goals and methods performed during the task, the amount of time without an active task, the number of perceived errors, and the number of task interruptions, all calibrated to the reporting scale through calibration parameters.

[10] EASE SCA and EASE RULEX-EM used the same workload mechanisms.

6. What is your account of multi-tasking?

ACT-R, EASE, and COGNET/iGEN all have similar accounts of multi-tasking based on the serial nature of some portion of their central cognitive processing mechanisms. Each architecture's cognitive multi-tasking mechanisms are based around goal or task switching. In ACT-R, cognitive behavior is goal-oriented with the current goal playing a key role in the selection of production rules from one cognitive cycle to the next. Only a single production rule can fire on any given cycle. Concurrent tasking can be accomplished by combining multiple goals into a single goal through extensive training. However, the architecture places limitations on both goal switching and goal combination capabilities. The incorporation of an explicit representation of sensitivity to visual onsets (see the section on *Perception* above) in ACT-R allowed for the possibility of task interruptions, and therefore increased reactivity in the model. This is an important milestone for the ACT-R group, because much of the cognitive modeling community had assumed that ACT-R's goal-focused orientation precluded the possibility of task interruptions, thereby limiting the utility of ACT-R as an architecture for modeling multi-tasking. That the addition of sensitivity to visual onsets made this possible serves as additional evidence for the modeling benefits to be gained by using an "embodied" cognitive architecture.

Unlike ACT-R, EASE is a parallel firing production rule system where each rule that matches fires on every production cycle. In EASE, rules register preferences for operators. It is these operators, representing tasks, which operate in serial and manipulate or transform goal states. By encoding arbitration and priority guidelines for operator selection into production rules, the system is able to represent cognitive task switching. Critical to the success of multitasking in EASE was the addition of a capability for task interruption, driven by perceptual input. Constraints on multi-tasking come largely from task instructions and strategies encoded in production rules or limitations in the perceptual or motor components of the architecture.

COGNET/iGEN was designed with multi-tasking as a primary component of the system. In COGNET, the cognitive, motor, and perceptual subsystems operate in parallel. Within the cognitive system, only a single procedural knowledge unit, or task, can be executed at one time. However, multiple tasks can simultaneously be active with differing states of completion. Changes to memory knowledge structures can facilitate the interruption, suspension, or execution of tasks. Multi-tasking is facilitated through parallel task activation and rapid task switching and execution. The COGNET/iGEN team added a separate knowledge type (metacognitive

knowledge), which in conjunction with declarative and procedural knowledge facilitates the ability of the model to multi-task on a strategy-driven basis. Alternative meta-cognitive strategies for multi-tasking are built into individual models by the developers via changes to the metacognitive knowledge. However, even without an explicit multi-tasking strategy, a model can still multi-task using iGEN's built-in meta-cognitive strategies. A metacognition module makes for an effective means of managing activity during multi-tasking, and is unquestionably a useful architectural component. One might question the theoretical parsimony of a separate metacognitive knowledge mechanism, but it is important to keep in mind that the COGNET mission is not one of theoretical improvement, but rather to create a behavior representation system with practical applicability in a wide variety of modeling contexts.

With its primary emphasis on cognition as a distributed state-change system, DCOG has inherent mechanisms facilitating multi-task performance. In DCOG, tasks consist of activities performed by software agents operating in parallel, and independent from one another. Behavior is coordinated, and constraints imposed, either through task sequencing occurring within an individual software agent, or through signals broadcast between agents. Without these constraints, tasks executed by different agents are all processed in parallel.

7. What is your account of categorization?

While there have been instance or exemplar based accounts of category learning in the literature (e.g., Kruschke, 1992; Love & Medin, 1998), all the AMBR models were either rule-based models, or hybrid models, combining both rule and instance-based learning.

Originally developed using an instance-based approach, the initial ACT-R model could not replicate the steep learning curve found in the Problem Type I data. The revised ACT-R model included the addition of six single-dimension production rules whose subsymbolic production utility values compete with each other and with the instance-based learning mechanism to try to succeed at the categorization task. In the Type I condition, one of these single-dimension categorization rules will always be successful and its utility value therefore dominates the others very quickly, leading to rapid improvements in categorization performance. Limits to instance-based category learning capabilities are based on memory decay and retrieval (e.g., partial matching) mechanisms.

The EASE SCA approach to categorization is a specific-to-general search for a production rule matching the instance. It first looks for a rule matching all features. If unsuccessful, it

ignores a feature and again attempts to find a match. This is performed until either a match is found or all features have been eliminated, resulting in a guess being generated. Learning occurs by constructing a new specialized production based on the production used to generate the response. The last feature unspecified by the old production is set to the value specified in the stimulus. In this manner, the system will eventually saturate, learning fully specified rules for each stimulus. This approach learned too quickly, relative to the human data, so the modeling team hypothesized participants were attending to additional features irrelevant to the task. These 'noise' features reduced performance to levels matching the human data.

The RULEX-EM model learns both instances and rules, both of which are represented as declarative memory structures. Like EASE SCA, RULEX-EM performs a specific-to-general search strategy, first attempting to recall instances, then two-feature rules, followed by single-feature rules. In the event a complementary single feature rule is found (i.e., the single feature value is the opposite of that found in the stimulus), the model responds with the complementary category. If no applicable rule is found, the model makes a guess. RULEX-EM is a learn-on-failure algorithm, with new instances or rules learned in response to failures. Existing rules and instances are rehearsed when successfully applied. Based on the memory mechanisms of ACT-R, items used frequently are more likely to be retrieved from memory, while unused items decay and are forgotten.

The COGNET/iGEN model also takes a rule-based approach to category learning. However, unlike EASE SCA or EASE RULEX-EM, COGNET/iGEN performs a general-to-specific search strategy; first attempting single-feature rules, followed by two-feature rules, and lastly three-feature rules or unique instances. Rules in COGNET/iGEN are specified only for a single category response; acceptance. If no applicable rules are found, the stimulus is categorized into the 'rejection' category. Rule learning occurs using a general-to-specific strategy. New single-feature rules are hypothesized if no existing rules match the stimulus. Incorrect rules are further specialized based on negative feedback, or removed if already fully specialized. Limits on the speed and power of learning are due to constraints placed on memory mechanisms. Only a limited number of rules are retained in short-term memory, with the likelihood of recalling a rule from short-term memory the product of rule complexity and the amount of rehearsal afforded the rule. Rules are maintained in short-term memory through rehearsal, and transferred to long-term memory based on memory load and rule complexity. Rules in long-term memory are always

available to the categorization process, and are not subject to decay, but can be deleted deliberately.

The DCOG model employs four different learning strategies (rote, emergent, deductive, and abductive). DCOG claims that humans learn by using strategies, and that subjects may shift learning strategies as they interact with their environment. They also claim that people have individual differences in their preferred learning strategy, in their patience with the learning process, and in their tolerance of exceptions. DCOG characterizes these individual differences with the parameters for mode, patience, and tolerance, which are called "personality" variables.

4.3.2 Summary

Table 11 summarizes the features found in the architectures and models involved in the AMBR project.

Table 11: Architecture Summary Table

	ACT-R	**COGNET/IGEN**	**D-COG**	**EASE**
Representation of Cognition	Modular architecture interacting through a central production system. Declarative memory is represented as structured chunks while procedural knowledge is represented in production rules	Flexible architecture with multiple representations for knowledge. Cognitive processing is serial with rapid attention switching based on changes in memory components.	Software agent architecture with agents executing parallel computational threads. Emphasis on strategies, which are a foundation for behaviors.	Match and fire production system with rules and declarative memory structures.
Modeling Method	Model construction emphasizing the power and constraints of the architecture	Use Cognitive Task Analysis to capture strategies from human experts. Development driven by task demands and constructed at appropriate level of granularity.	Perform task analysis to develop models specialized for work/task.	Emphasis on listening to the architecture and building models within the constraints imposed by the architecture. Some use of Cognitive Task Analysis.
Parameters	Work within the constraints of the architecture. Use default parameter values where available. Knowledge structures the greatest source of degrees of freedom for the model.	Core architecture has no parameters. Makes extensive use of parameterized micromodels to fit empirical data.	Parameters reflect different learning strategies and personality factors. Strategies and parameter values selected based on empirical data and task analysis.	Knowledge structures considered the primary "parameter". Reformulation of knowledge has large impact on performance. Default ACT-R parameter values used for memory mechanisms.
Individual Differences	Differences due to variations in knowledge structures, architectural parameters, and stochasticity built into system components.	Stochasticity built into micromodel components and feature selection mechanisms. Dominant differences based on knowledge and task strategy differences, although only one strategy was implemented in AMBR.	Large emphasis on multiple learning strategies. Twelve individual models were developed, each reflecting different learning strategies and personality factors.	The SCA model developed twelve different model instances with different learning strategies and parameter values. RULEX-EM was a normative model and did not account for individual differences.
Workload	Scaled ratio of time spent on critical task goals to total time on task, combined with a weighted measure of errors made.	Detailed workload assessment across all six workload dimensions using a complex set of factors.	The presence of up to eleven factors were weighted and combined into a single measure of workload.	Scaled sum of the perceived amount of work required. Different types of work were differentially weighted.

	ACT-R	COGNET/IGEN	D-COG	EASE
Multi-tasking	Cognitively controlled goal switching. Concurrent tasking is possible by combining goal representations	Parallel processing across motor, perceptual, and cognitive components. Within the cognitive system multiple tasks can be active, but only one is executing at one time.	Inherently multi-tasking architecture. Tasks are activities performed by software agents operating in parallel. Constraints occur via task sequencing within an agent or by waiting on event signals.	Multi-tasking is implemented as task switching between task operators. Operator/task preferences encoded in production rules.
Categorization	Hybrid approach combining instance-based memory retrieval with rule-based productions for single-feature rules. Performance limitations based on memory decay and retrieval mechanisms, as well as production utility learning mechanism.	General-to-specific rule-based hypothesis testing. Constraints based on short-term memory limitations and knowledge transfer to long-term memory.	Four strategies for learning and mechanisms for switching between them; including feature and exemplar learning and hypothesis testing. Categorization implemented as a spreading activation model.	SCA is a specific-to-general search for production rules matching the instance. Learning occurs by production specialization. RULEX-EM is a hybrid model using both exemplars and rules.

Even with their diverse origins and varied length of existence, there are a number of commonalities and points of convergence among the models. As aspiring unifying and integrative cognitive architectures, ACT-R and EASE share the greatest amount of architectural commonality. Both are built on production systems with similar cognitive representations – with EASE even utilizing some of ACT-R's memory mechanism. Both EASE and ACT-R primarily use task or goal switching to model multi-tasking performance. Conversely, DCOG and COGNET/iGEN were designed from the onset for parallel processing, with inherent multi-taking capabilities. ACT-R and EASE are also similar in that both architectures emphasize the importance of working within architectural constraints while building models, and taking an architecturally centered approach to model development. Even with these constraints, both architectures provide a significant degree of freedom to develop knowledge representations across diverse modeling tasks. Conversely, DCOG and COGNET/iGEN emphasize a more task oriented approach to model development. For these architectures, task analysis is used to provide the foundation on which to develop models specialized for a particular task.

While the pair-wise points of divergence between the architectures and the models developed with them are numerous, there are a few points of divergence that are particularly noteworthy. First, each model's implementation of workload varied considerably, with each model using a wide variety of factors and processes for calculating workload. Secondly, DCOG's architectural framework as a set of parallel, interacting software agents is unique among the architectures. Lastly, COGNET/iGEN's lack of core architectural parameters and consequent use of parameterized micromodels to fit empirical results was also unique among the architectures.

4.4 References

Agresti, A. (2002). *Categorical Data Analysis, second edition*. Wiley, New York, NY.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (1993). *The rules of the mind*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Baker, R.S., Corbett, A.T., Koedinger, K.R. (2003) *Statistical Techniques For Comparing ACT-R Models of Cognitive Performance*. In Proceedings of the 10th Annual ACT-R Workshop, 129-134.

Bozdogan, H. (2000). Akaike's Information Criterion and recent developments in information complexity. *Journal of Mathematical Psychology, 44*, 69-91.

Busemeyer, J. R., & Wang, Y.-M. (2000). Model comparisons and model selections based on generalization criterion methodology. *Journal of Mathematical Psychology, 44*, 171-189.

Gluck, K. A., & Pew, R. W., (Eds.) (in press). *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation*. Mahwah, NJ: Lawrence Erlbaum Associates.

Grünwald, P. D. (2000). Model Selection based on Minimum Description Length, *Journal of Mathematical Psychology, 44*, 133-152..

Gureckis, T. M., & Love, B. C. (2003). Towards a unified account of supervised and unsupervised learning. *Journal of Experimental and Theoretical Artificial Intelligence, 15*, 1-24.

Hart, S., & Staveland, L. (1988). Development of the NASA-TLX: Results of empirical and theoretical research. In P. Hancock and N. Meshkati (Eds.), *Human mental workload*. Amsterdam: North-Holland, 139-184.

Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation, *AI Magazine, 20*, 27-41.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*, 391-438.

Kieras, D. E., & Meyer, D. E. (2000). The role of cognitive task analysis in the application of predictive models of human performance. In J. M. C. Schraagen, S. E. Chipman, & V. L. Shalin (Eds.), *Cognitive task analysis*. Mahwah, NJ: Lawrence Erlbaum Associates.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review, 99*, 22-44.

Love, B. C. & Medin, D. L. (1998). SUSTAIN: A Model of Human Category Learning. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 671-676.

Miller, C. S., & Laird, J. E. (1996). Accounting for graded performance within a discrete search framework. *Cognitive Science, 20*(4), 499-537.

Myung, I. J. (2000). The importance of complexity in model selection. *Journal of Mathematical Psychology, 44*, 190-204.

Nosofsky, R.M., Gluck, M., Palmeri, T.J., McKinley, S.C., & Glauthier, P. (1994a). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory & Cognition, 22*, 352-369.

Nosofsky, R.M., Palmeri, T.J., & McKinley, S.C. (1994b). Rule-plus-exception model of classification learning. *Psychological Review, 101*, 53-79.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Cambridge University Press.

Pitt, M. A., Myung, I. J., & Zhang, S. (2002). Toward a method of selecting among computational models of cognition. *Psychological Review, 109*, 472-491.

Roberts, S. & Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing. *Psychological Review, 107*, 358-367.

Roberts, S. & Pashler, H. (2002). Reply to Rodgers and Rowe (2002). *Psychological Review, 109*, 605-607.

Rodgers, J. L., & Rowe, D. C. (2002a). Theory development should begin (but not end) with good empirical fits: A comment on Roberts and Pashler (2000). *Psychological Review, 109*, 599-603.

Rodgers, J. L., & Rowe, D. C. (2002b). Postscript: Theory development should not end (but always begins) with good empirical fits: Response to Roberts and Pashler's (2002) reply. *Psychological Review, 109*, 603-604.

Schunn, C. D. & Wallach, D. (2001). Evaluating goodness-of-fit in comparisons of models to data. Online manuscript. http://www.lrdc.pitt.edu/schunn/gof/index.html

Vidulich, M.A. & Tsang, P.S. (1986). Collecting NASA Workload Ratings: A Paper-and-pencil Package (Version 2.1), Working Paper. Moffett Field, CA: NASA Ames Research Center.

Wasserman, L. (2000). Bayesian model selection and model averaging. *Journal of Mathematical Psychology, 44*, 92-107.

5. Accomplishments, Challenges, and Future Directions for Human Behavior Representation

(Richard W. Pew, Kevin A. Gluck, Stephen Deutsch)

5.1 Summary of Accomplishments

In Chapter 1, (this report), Gluck, Pew, and Young described three goals for the AMBR Model Comparison: (1) to advance the state-of-the-science in human behavior representation (HBR), (2) to develop HBR models that are relevant to the Department of Defense mission, and (3) to make all of the research tasks, human behavior models, and human process and outcome data available to the public. As evidence of progress on the first two goals, in this book we have presented an exemplary set of models and the modeling architectures in which they were built. In Experiment 1 the models pushed the frontiers in the representation of multi-tasking in HBR architectures. In Experiment 2 we stimulated the incorporation of category learning into architectures that previously did not have this capability. These accomplishments are certainly a contribution both to the state-of-the-science and to the development of more capable models to meet DoD HBR needs.

The book (Gluck and Pew, in press) and the accompanying CD represent the accomplishment of Goal 3. Early in the project we opened a website and BBN made available runnable copies of the software supporting the project. On the CD, in addition to the runnable simulation software, we have included data files and material from each model developer documenting their model implementations.

Beyond progress toward the primary goals, the project has also confirmed that it is feasible to conduct comparisons among models at this level of complexity on a common problem and that doing so is a useful way to assess current capabilities and stimulate further advancements and cross-fertilization among proponents of the various architectures and modeling approaches. The comparison paradigm is an effective way to advance the field. In the course of conducting the comparisons we learned a lot and also identified a number of issues that need to be addressed to enhance the contribution that such comparisons can make. These will be addressed in subsequent sections.

5.2 Challenges to the Conduct of Model Comparisons

We have been advocates of the comparison approach to pushing the frontiers of models for some time. When it came to actually accomplishing it we had to address a number of issues.

5.2.1 Choice of Domain and Task

At the outset, we had the challenge of selecting the human performance tasks to be modeled. Even when we had agreed on the thrust of the comparisons (multi-tasking and category learning), there were difficult tradeoffs to be considered in choosing a task context. We could select a task that was of practical interest, realistic complexity and required highly trained operators to be our participants, such as a high-fidelity simulation. Or we could select a task that was highly abstracted, like a traditional laboratory task from experimental psychology that anyone could be expected to learn in a very short time and that would isolate the cognitive phenomena of interest.

Clearly the first alternative has greater practical significance and is more challenging from a modeling perspective. However, it would have required extensive knowledge acquisition on the part of each development team, an investment that would detract from the time and effort that could be put into the actual coding of models. The BBN Moderator team could have supplied that knowledge, but we were concerned that knowledge supplied by a third party might not be sufficient. An overlay on this debate was whether we should require the developers to model experienced operators or novice operators. There were strong arguments against modeling novices doing highly complex, real-world tasks, because the likely variability they would produce in the data would mask the behaviors we were trying to measure.

Using a high-fidelity task also would have had implications for the moderator team. We had limited resources for collecting data. Either we would have had to identify and recruit experienced (and expensive) operators from the domain under study to employ as participants, or we would have to invest in a very extensive period of training (which also is expensive).

As a compromise between a high-fidelity simulation and an abstract laboratory task, we opted to use an abstracted version of an air traffic control task. The task is probably not as representative of multiple task management or category learning requirements as we could have achieved with a more realistic task, but we obtained stable data from novice human participants in four-hour sessions and the modelers were able to develop the requisite knowledge based on their own experience or that of a small set of previously untrained subjects.

5.2.2 What Human Data To Collect

A major goal of our approach was to collect data from humans and from the models for purposes of making model-to-human and model-to-model comparisons and to give the modelers human data to allow them to tune their models to the details of the task requirements. As has been amply

demonstrated in Chapter 4 (this report), this is not as straightforward as it might seem. There is a need to satisfy multiple criteria for choosing the data as well as for comparing results. First we wanted to collect data sets that all the model developers would find useful for tuning their models. Since each model began from very different theoretical bases and software infrastructures, the data potentially useful for one model may not be useful for any other. For example two of the models would have found eye movement records of the human subjects useful data, however at least one of the models did not make any assumptions about the details of eye movements at all.

Second, we took as a requirement that we be able to obtain the same data from both the human subjects and from the models. This is more constraining on the humans than on the models. One could imagine a range of data that can be collected from the models that would not be available from humans, such as the average number of tasks queued waiting for execution. This led us to focus on the more or less standard measures -- aggregate measures of observable outcome performance, such as task completion time and number of errors. We collected data on response times for the most elemental task decomposition elements for which we could reliably identify both a stimulus event and a response event. We also required the model developers to provide an estimate of workload level derived from their theory and their models' performance, and we compared that with human participant subjective workload data as measured by the NASA TLX instrument (Hart & Staveland, 1988; Vidulich & Tsang, 1986). We found this a useful addition to the direct performance measures and it challenged the developers to think about how they would represent workload. In addition we provided a trace of the time history of every action of each scenario for each subject in case the model developers wished to analyze it to obtain some other parameter or index. These traces also made it possible for the developers to rerun a trial as performed by a participant and watch the resultant activity on the ATC-like displays

Third, we wanted the data to be useful for discriminating among the features of various models. But for the same reasons, namely a lack of commonality in the decomposition of either human performance or of the tasks to be performed, we did not figure out how to specify measurements at the level of model features that would be universally comparable, even if we dropped the constraint that the same data had to be collected from humans. Instead we settled for asking each of the model developers to answer a common set of questions, the results of which

are reported at the end of each of the model chapters (Chapters 4-7, Gluck and Pew, in press) and summarized in Chapter 4 (this report).

Finally, we wanted to collect data that would challenge the predictive capabilities of the models. To do so required that some of the data from human subjects be given to the developers to support tuning their models, but that a condition be added to the experiment that was not announced to the developers until after the models were declared complete. There was a real challenge to devising this additional condition. It could not be so different as to create a new task for which the developers had not prepared their model, but it could not be so similar that it did not represent a stretch for the models. In Experiment 1 we created a second set of scenarios that were substantially similar to the original ones, except that the arrival times and locations of the incoming planes requiring control were changed. This proved to be so similar to the model development scenarios that it did not represent a challenge at all. In Experiment 2 we created a categorization transfer condition wherein the subjects were to respond to new specific stimuli that were either interpolated among or extrapolated from the original ones. For these stimuli neither the human subjects nor the models had previous exposure. These conditions proved to be so difficult that none of the models were able to predict the behavior of the human subjects successfully. Nevertheless this proved to be a useful condition to introduce because, when given an opportunity to revise their models to do better on this transfer condition, a great deal was learned through an understanding of the specific nature of the changes that were required to make the models perform better.

5.2.3 Whether to Compare or Compete the Models

The large number of human behavior representation architectures available for use today (see Chapter 1, Table 1, this report) understandably leads to the common speculation that certain of these architectures are better than others, or at the very least, certain of them should be better at representing particular human abilities or behaviors. Such speculation results in the desire for competitions to help decide which is the best architecture to use. It was exactly this sort of interest that motivated an orientation toward competition early in the AMBR Model Comparison. However, as the project began to take shape, it became clear that selecting the HBR system that is objectively the *best* for representing human behavior is not an achievable objective.

There are three reasons for this. The most obvious is that we did not have the resources available to have every available HBR architecture participate. At best we could only hope to

provide a rank ordering among those architectures participating in the project. A second reason is that all current HBR architectures are moving targets. They are constantly under development and subject to modification, in order to expand their range of application and/or level of psychological validity. Therefore, any conclusion regarding the rank ordering of the architectures would be a fleeting conclusion – true only until the developers of the various systems had improved on any deficiencies revealed in the course of the project. The third reason is that each alternative is likely to have its own strengths and weaknesses, and choosing among those strengths and weaknesses adds a layer of subjective valuation that marginalizes any claims regarding which of the architectures is "the best." For these reasons, a rank ordering of architectures did not seem to be the appropriate goal.

Some consideration was then given to the possibility of using the project as an opportunity to compare the characteristics and capabilities of different HBR architectures that make it possible (or impossible) for them to represent the human capabilities of interest in the project. The proliferation of human representation systems in the modeling and simulation community has prompted others to adopt this goal before (Anderson and Lebiere, 1998, 2003; Johnson, 1997; Jones, 1996; Morrison, 2003; Pew & Mavor, 1998; Ritter et al., 2003), and it certainly is reasonable to think that we could follow suit. One positive aspect of this approach is that it shifts the emphasis somewhat from the brand names to the (more important) underlying architectural characteristics.

Architectural characteristics and capabilities are not the full story, however. Recently the point has been made that modeling style, or *idiom*, is at least as important to the success of a model as are the underlying architectural characteristics (Kieras & Meyer, 2000; Lallement & John, 1998). This means that any particular model's ability to simulate human behavior is a function not only of what the architecture allows the modeler to do, but also of how the modeler uses the architecture. Modeling style effectively becomes a confound in any effort to compete across architectures.

An additional consideration is that the best way to objectively demonstrate the relative utility of HBR architectures is to use them to build models. These architectures make predictions about human performance only through the models that are developed with them, which means that any competition really would be among models, not architectures. This suggests a focus on the models that are developed rather than the characteristics of the architectures which support their

development, and a consequence is the requirement to develop human behavior process models as part of the project.

Once it was clear that the process models themselves were to be the focus of the project, there still was the issue of deciding whether the project was to be a model *competition* or a model *comparison*. This is a distinction with implications we didn't fully appreciate early on. The two terms were used interchangeably, which created some confusion regarding the goal of the project. A competition implies the goal is to identify a winner. However, several concerns led us to the conclusion that a model competition was not what we wanted. First, an infinite number of models can be developed to describe or predict human behavior, some or all of which might do an equally good job of accounting for that behavior. Anderson (1993) referred to this as the "uniqueness" problem. To complicate matters further, there is no guarantee that the models created for such a competition are necessarily the best possible models that could have been created with those architectures. Therefore, any rank ordering that is based on modeling results must be considered tentative, and arguably could be considered misleading. Third, "winning" depends on one or more evaluation criteria, which raises the issue of what criteria to use. Candidate criteria might include empirical model fits to data, the parsimony with which the model represents the behavior, amount of re-use of knowledge, design principles or parameters from previous models, usability, interpretability, robustness, or development cost. Once the criteria are chosen, of course, decisions must be made regarding relative weighting. Models are developed for different purposes, with architectures created from various underlying motivations. These different architectural motivations and model purposes make it very difficult to get people to agree on what is important in HBR models, which makes it hard to reach consensus on the weighting of evaluation criteria. Finally, there was the pragmatic issue: there was nothing to win – no follow-on contract or cash prize. Thus, although identifying a "winning" model is an attractive idea on the surface, the uniqueness problem and debates regarding model assessment methodology and what should be valued in HBR models make it a problematic and potentially empty objective.

In the end, the decision was made to organize the project as a model *comparison*. Just as in the case of architectural comparisons, one benefit of this approach is that it emphasizes mechanism, as opposed to name-branding. It is more important to gain insight into the representational assumptions and processing mechanisms that enable the representation of human

behavior, than it is to establish that a particular brand-name architecture has a better fit to a sample of data than does another architecture. This does not mean goodness-of-fit is irrelevant in a model comparison, since a model's ability to predict or explain some aspects of human behavior is critical in the development of HBR models, and a useful means of assessing this is through a measure of fit. Nevertheless, the ultimate goal is to shift the emphasis away from the name brand, and toward the underlying assumptions and mechanisms that produce certain predictions. Another advantage of the comparison approach is that it promotes communication among modelers using different architectures, which in turn encourages architectural improvements. This is good for the modeling and simulation community, as it results in improved HBR capabilities.

Some might still argue (and they have, in both public and private conversations with the authors) that the AMBR Project should have pursued some form of head-to-head competition among the models in order to determine which is the "most correct." However, Estes (2002) points out that although this argument seems sensible on the surface, it becomes problematic upon closer inspection. One problem is that models created by different people using different modeling architectures will differ in numerous ways and make it extremely difficult, if not impossible, to determine which are the necessary assumptions that allow for successful prediction of the data. Another problem is that as model complexity increases, it becomes more likely that the one that "wins" will simply be the model that is favored by the sampling error in those data. This tells us less about the necessary human behavior representation requirements for predicting specific behavioral phenomena than it does which of the models happened to win the statistical lottery.

5.2.4 Summary

These challenges and others will be faced by anyone attempting future large-scale model comparisons, and many of them also are faced daily by people involved in the cognitive modeling and human behavior representation communities. We will use the remainder of this chapter to offer two forms of guidance for further advancement in HBR modeling: (1) programmatic guidance for development of future model comparisons and (2) science and technology guidance regarding specific directions for improvement in the theory and practice of modeling.

5.3 Guidance for Future Model Comparisons

As described in Chapter 1, this report, the model comparison process employed in this project began with selecting the driving goals for improved models, identifying a suitable task domain and creating a simulation of it that could be operated either by a human-in-the-loop or by a model. Then a workshop was held to exchange ideas, interfacing requirements and constraints among the Moderator and Developer team members so that everyone was on the same page. Following this step the Moderator team collected and disseminated human performance data for the task and the modeling teams generated models that sought to replicate the data. Finally an expert panel was convened with the entire team to review the data and contrast and compare the models and the results were shared with the scientific community. On the basis of comments made after the completion of Experiment 1, it was recognized that the simulation could have been more complete before the workshop and that the expert panel should have been involved earlier. For Experiment 2 some progress was made in accomplishing both of these goals, but it was concluded that still more needed to be done to get the developers and the Expert Panel on board.

5.3.1 More Modeler Input

Although for Experiment 2 the simulation was more complete before the workshop, we believe that much more collaborative effort should be invested among the model developers and moderator team, particularly after the task is selected and the task simulation has been created and signed off. At this point the developers are the only ones who know their models well enough to suggest the level at which to define measures that might discriminate among the model features, and yet be applicable across models. Initially the Developers Workshop was held at a time when the task was not signed off and it focused mostly on what parameters had to be exchanged with the simulation in order for the model and simulation to execute together. The moderator team dictated what data the modelers were to provide rather than engaging in an exchange that might have led to a more creative definition of performance measures.

5.3.2 Get Objective Expert Guidance

We also believe that the use of an expert panel was an extremely beneficial feature of the comparison process. They brought a very high level of knowledge and experience to the project. Even after they were brought in earlier in Experiment 2, they had great difficulty understanding the models in sufficient depth to consider comparative strengths and weaknesses realistically. In

future such comparisons it is critical that the panel be identified at the beginning of the project and brought into the process early, certainly by the time of the Developers Workshop.

5.3.3 Focus on Prediction

The use of a predictive phase in the model development and revision cycle in which the modelers were required to predict the results of the transfer task provided another mechanism by which to compare the models, and provided a significant challenge to the modelers, illustrating the fragility that most models exhibit at the boundaries of their intended performance envelopes. We suggest that this strategy of evaluating the predictive capabilities of a model can be quite valuable and should be considered as part of the evaluation of any model or modeling architecture.

5.3.4 Just Do It

In our opinion the importance of undertaking such comparisons outweighs the challenges that we have encountered in accomplishing this one. It is a productive way to push the frontiers in terms of model architecture development as well as our understanding of stable, productive methodologies for moving from architecture to detailed, robust human performance models.

5.4 Needed Improvements in the Theory and Practice of Modeling

Significant scientific and technological progress is needed on a variety of fronts. First, we need to continue to improve our theory and practice for building robust models. Second, we need to strengthen the research base concerning human integrative behavior on which robustness can be built. Third, we need to continue to improve the validation process, including the metrics against which we evaluate. Fourth, an important contribution to validation would be to reduce the opacity of models through improved inspectability and interpretability. And finally, as we improve the theory and practice, we need to develop and assess better methods for improving cost-effectiveness with respect to the ways in which the models are created and used. In subsequent paragraphs we will address each of these requirements.

5.4.1 Improving Robustness

The quote, "Robustness is the ability of a system to maintain function even with changes in internal structure or external environment"[11] provides a good working definition of robustness where the "systems" are our human performance models. Making models more robust in this sense is a high priority goal for future development that should be funded both at the application level and at the basic research level. Researchers are at work on several paths by which to improve model robustness. The challenge will certainly stimulate ideas for new initiatives. For the present, we provide an outline of one strategy composed of elements at three levels of complexity, each with the potential to contribute to model robustness. The focus of this strategy is greater tactical variability.

The methodological norm for all architecture-based model development today is that developers, using their preferred cognitive architecture, start from a task definition and environment description, elaborate the contingent elements of the task, the decision points, and attempt to anticipate different potential outcomes. They then program the models to meet their challenges, conditional on the series of real-time cues that define the context in which they operate. We will refer to these sequential task executions as "threads" because they may span multiple levels in the task hierarchy and may involve execution of multiple productions or procedures. What developers don't do often enough is to think tactically about the multiple ways the same threads under the same constraints could be performed or how slight variants in the constraints or cues might lead to different behaviors and outcomes. Developers tend not to represent a large number of tactical variants because of the added complexity and associated costs in coding time. The small number of variants represented is a significant concern because the availability of an assortment of tactics for accomplishing any given task has significant advantages over the limited repertoire more typical of most models developed today. These kinds of variations contribute in several ways. They are required in order to represent inter- and intra-individual tactical variation from execution to execution. In training simulations, such variability makes it more difficult for trainees to "game" the system by easily anticipating synthetic force actions. Additionally, inter-individual variation is needed to represent cultural differences and

---

[11] http://discuss.santafe.edu/robustness/ ; Definition 2 (SFI RS-2001-009 Posted 10-22-01) [12] If it is a training simulation the usefulness is captured in measures of training effectiveness. If it is an evaluation associated with system acquisition, usefulness rests in the ability to discriminate real differences between alternative designs

personality differences. All of these kinds of variations contribute to robustness in a general sense, but do not address the ultimate problem of dealing with unanticipated events.

There are three different levels of sophistication in the methods by which tactical variation can be introduced and monitored. At the simplest level, at significant decision nodes, the developer ought to anticipate the reasonable alternate ways that a modeled human could accomplish the given task. There may be alternate means to achieve the same outcome or an appropriate alternate outcome that might be pursued. The decision on the outcome to be achieved and the means to achieve it can be based on contextual cues or even simple stochastic selection. This kind of variation can be accomplished in just about any architecture, but it takes significant time and effort.

At a more challenging level of sophistication, we can add an adaptive selection mechanism so the model learns from the choices that it makes at the decision points where it chooses among alternate outcomes and the means to achieve those outcomes. The advantage here is that once the tactical variants are coded in, the model can learn to select among them to improve probability of success. Three elements are necessary to support the learned selection:

(1) A rich array of executable behaviors, not all of which lead to the same result, but which better represent the range of actions a human might take.
(2) A real-time selection mechanism that chooses among possible behaviors appropriate in the present situation to achieve the goals of the particular thread.
(3) A real-time learning mechanism supported by performance indices that measure success or failure and credit the appropriate thread decision relative to the situation in which it was made.

Ideally, the selection and learning would take place internal to the cognitive mechanisms that make up the model. That is, they might involve task activities such as planning or workload estimation, that will change the course of action well before the final procedure or production is executed to produce the behavior. This places demands on the performance indices to represent success at appropriate intermediate as well as final levels in the completion of the threads. The most challenging part of this approach is selecting the threads that can benefit from this treatment, identifying the appropriate decision nodes, and creating and scoring the effectiveness measures that are appropriate to each thread or context. In other words, the challenging scientific issue is how to handle the credit and blame assignment.

The most difficult challenge lies in adapting existing behaviors or creating new behaviors to address unanticipated situations. If we wish to genuinely improve model robustness, that is, the ability of the models to continue to execute reasonable behaviors in contexts that were not anticipated by the developers, the models will need to do what people routinely do every day – combine elements of old threads in new ways and to create new threads to meet the situations at hand. This is the most significant scientific challenge facing the human behavior representation community; one that is seldom acknowledged. It is a capability largely untouched by today's architectures. Some believe that it can be achieved within one or more of the existing architectures, while others believe an entirely different and as-yet-undeveloped architecture will be necessary. It is a topic worthy of significant research investment.

5.4.2 Improving our Understanding of Integrative Behavior

By integrative behavior we mean the aspects of skills that transcend perception, cognition and motor performance. In the last several years, as cognitive psychologists have moved away from the componential view of information processing and toward more integrative views based in neuroscience, there has been growing interest in understanding the coordinative and integrative functions that bring together the wide range of basic human abilities. Theories of workload and multitasking fall into this category. People monitor their own ongoing performance. People dynamically prioritize, schedule, and re-plan. Somehow they know when they are running out of time or falling short of succeeding and have the means to try something different. How are these objectives accomplished? People's cognitive skills, as well as their motor skills, improve with practice. Novices are different from experts. All of these characteristics of real human performance reflect something more than basic perceptual-motor skills. Advancing this understanding and manifesting it in models represents a second requirement for more robust models. Eva Hudlicka's work on modeling emotion (Hudlicka, 2002) represents a start toward identifying and representing the impact of emotion at the level of cognitive mechanisms, and we see the integration of emotion into HBR architectures as an important step.

5.4.3 Improving Validation

At this point we should pause and take stock of exactly what we should be expecting of our models. It is often argued that the *sine qua non* of human behavior representation is for the models to behave exactly as humans do. The authors' experience suggests otherwise. We believe the criteria for success of models should be usefulness, not veridical representation of humans.

We must remember that models are only approximations. Is it not true that one of the reasons that we believe models are useful is because they are simpler to build and, once built, to run, than actually to collect the corresponding data on humans?

What makes a model useful? First and foremost, it should serve the purpose for which it was designed. In other places these purposes have been defined to include training, evaluation at the level of conceptual design, supporting system acquisition decisions and in test and evaluation. Campbell and Bolton (Chapter 10, Gluck and Pew, in press) refer to this characteristic of models as Application Validity. In that chapter we have read about the limitations in the capabilities for validating models of the complexity and comprehensiveness of HBR models. It is worrisome that current DoD HBRs rarely receive anything more than cursory face validation, if that. Models whose results are to be relied on for training or system effectiveness assessment really need to be accredited for the purposes to which they will be applied, but quantitative accreditation guidelines do not currently exist. Much further creative work must be accomplished and we need to work toward a consensus for what classes of models or perhaps what aspects of models we expect to be able to validate quantitatively through goodness-of-fit to data or careful systematic qualitative analysis to assure they are reflecting what people really do, and what classes can only be validated through a usefulness demonstration. Usefulness is a fine criterion, and it is influenced by the quality of our HBRs, but it also depends on many other aspects of a simulation.[12] When it fails, it is very difficult to pinpoint the source of the failure. Even if it can be attributed to the HBR, without more detailed validation data, it is difficult to identify what it was about the HBR that needs improvement.

As we move toward ever more complex models and toward models that learn and adapt over time, validation can only be said to be meaningful at the level of representing the behavior of individuals, not aggregates of non-homogeneous individual performances (Gobet and Ritter, 2000). In the AMBR category learning experiments, we found it helpful to classify subsets of participants in terms of their strategies (i.e. those that cited using rules and those that didn't) and to evaluate the models' success at mimicking the corresponding strategies. The DCOG model explored the possibility of parameterizing personality differences that might lead to different behavior, or even different strategies (see Eggleston, McCreight, & Young, Chapter 6, Gluck and Pew, in press), but no other models that we are aware of have attempted to characterize different strategies and branch on the basis of some individual difference characteristic to follow different

strategies in accomplishing a task. We need to generalize and formalize the procedures for evaluating models of this type.

5.4.4 Establishing the Necessity of Architectural and Model Characteristics

We have already mentioned several times the importance of emphasizing the identification of computational mechanisms capable of reproducing desired human behaviors. If we wish to be diagnostic about where or why an HBR failed or succeeded, we need to further mature our methods for validating the characteristics of models that claim to represent the unobservable properties of cognitive mechanisms. In AMBR, the emphasis has been on encouraging development of models that are *sufficient* for postdicting and predicting human performance data. Establishing the *necessity* of architectural and model characteristics has not been a goal. Estes (2002) makes clear the distinction between necessity and sufficiency in model testing, and he describes this distinction in the context of the search for a correct (i.e., appropriate) model to account for experimental data. Appropriate is defined as a model that "... is necessary and sufficient for prediction of the data" (p. 5). If a model's predictions are correct, one can conclude that the model is sufficient for predicting the data. One can not conclude, however, that the assumptions of the model are necessary. Necessity can only be demonstrated by modifying the model's assumptions and showing that the modified version no longer predicts the data. If this can be shown, then the assumptions and processes built into the model are, in fact, necessary and sufficient for predicting the data, and therefore they comprise an appropriate model. Estes writes:

> ... an improved strategy is, at each step, to compare a reference model with an alternative version of the same model that differs from it with respect only to inclusion or exclusion of a single parameter or process. One gains information, not only about the sufficiency of the reference model for predicting the test data, but also about the contribution of the component whose exclusion leads to (relative) disconfirmation of the model. (p. 8)

This form of comparative testing of models is to be done *within an architecture*, varying only one free parameter at a time. Due to budget and time constraints, we have not adopted this approach in AMBR. However, we agree with Estes that this is a productive approach for testing the necessity of the components of computational process models, and perhaps it will be possible to follow this recommendation in future research that builds on the existing AMBR models.

5.4.5 Improving Inspectability and Interpretability

Opaqueness was an issue with the models in AMBR, as with all complex computational process• models. Diller, Gluck, Tenney and Godfrey (Chapter 4, this report) refer to this as model interpretability and Love (Chapter 9, Gluck and Pew, in press) refers to this as transpicuity. When it came time to compare the models, the developers were given two to four hours to present their models before a panel of experts who were generally familiar with HBR and the panel's conclusion was they did not learn enough in four hours to really evaluate the models. We need to develop means to make the internal performance of HBR models more transparent. This capability is useful for developers to support model debugging, it is needed to support validation and it helps users to understand just what a model is doing. One should never make use of software without an understanding of how it works. This point is often cited with respect to statistical programs, but is equally applicable to decision support and to models. The EASE model described by Chong and Wray (Chapter 7, Gluck and Pew, in press) includes a run-time interface that provides a visualization of where the model's eyes are looking, what buttons are being pushed and what perceptual/cognitive process is operating at each point in time. In an earlier model development effort, Nichael Cramer and Stephen Deutsch introduced a run-time interface with "Kate," a simple stick figure seated at a workplace whose eyes and hands moved tracing the model's execution of the task (Deutsch, Cramer and Feehrer, 1994). These features were very helpful in gaining confidence in how the model worked. Much more can be done to make a model's functionality transparent.

5.4.6 Improving Cost-Effectiveness

As was cited above, to be cost-effective a model should accomplish the intended purpose "well enough" and with less effort than one or more candidate alternative methods for accomplishing the same purpose. In the context of DoD applications cost-effectiveness of HBR models is established when the total lifecycle cost of a system with the HBR model can be shown to be less than the total lifecycle cost of a system without the HBR model.

What we mean by accomplishing the task "well enough" is a challenge. Just as in engineering design, there are literally hundreds of decisions and choices that are made in assembling a human-like model. Only a subset of them really makes a difference in the behavior that is relevant to the context and task of interest. The challenge, maybe an insurmountable one, is to determine those decisions that really matter and put the development effort into accomplishing

them well. One useful direction for further work might be to draw on the experience of successful models and model developers to catalog those features of working models that have turned out to be the most critical for achieving useful models. Since each modeling approach involves different decompositions of the requirements, there may be limited cross-fertilization possible, but the AMBR program has provided several examples where features have migrated from one model architecture to another (see Chapter 4, this report).

One way to promote cost-effectiveness is to invest in one model that can serve multiple purposes. One can create a model infrastructure and modular components that can be assembled as needed. This is the approach that was adopted by Robert Wherry and his colleagues in the original conception of the Human Operator Simulator (HOS) (Lane, Strieb, Glenn and Sherry (1980). It is similar to the approach used by those who create HBR architectures. Architectures exist along continua of completeness, scope and generality. At one extreme, illustrated by MicroSaint. (Micro Analysis and Design, 2004), they are a programming infrastructure in which to create models. At the other extreme, illustrated by ACT-R, they embody relatively complete theories of human performance and, in principle, require only adding the domain knowledge and constraints associated with the specific task context.

As presented in Chapter 1, Table 1, this report there are already more than two dozen alternative architectures advertised in various stages of completeness. Some have suggested that this is a bad thing and rather than continue to invest in such a broad range of architectures, the U.S. should concentrate its resources on just a few. It is our view that it is premature to settle on a small set of modeling approaches. The range of needs is broad. Different modeling architectures and approaches are needed because this is not a domain where one-size-fits-all. There will be opportunities for cross-fertilization from one approach to another and even the potential for aggregation across architectures, such as the integration of concepts from EPIC into ACT-R to produce ACT-R/PM (Byrne & Anderson, 1998) or the work of Chong to integrate EPIC with Soar to study learning of simple perceptual motor actions (Chong & Laird, 1997).

We found the development paradigm we evolved in the AMBR program to be one way to promote cost-effectiveness, especially since human performance data were available for the same tasks. Given an architecture, the method began with an understanding of the task requirements. Next, the context dependent features were added to the model. Then the model was run as a predictor of the human data before revising or tuning it to the specifics of the data. That way one

promotes an understanding of the capacities and limitations of the basic model and can more intelligently determine what is needed to improve it. In the process one gains an understanding of how robust it is in those cases for which data exist and this will provide a forecast of how robust it is likely to be in situations for which data are not available. In AMBR we were only able to accomplish this with the transfer condition of the category learning task (See Chapter 4, this report), however the inadequacies of the models before they were revised was striking and very revealing.

5.5 Concluding Thoughts

The message in this discussion for those funding and performing R&D on HBRs is that at the current state of development, resources need to be allocated not just to building the models, but also (1) to collecting the knowledge and human performance data needed to make them function realistically, (2) to iteratively conduct formative and summative evaluations to assure robustness, usefulness and validity and (3) to continue to support new science leading to breakthroughs in concepts for improved architectures and more robust models. If the military services intend to continue to increase their reliance on human behavior representation to improve the cost-effectiveness of training, system acquisition and decision-support, and there is every indication that they do, then it is short-sighted to support only the specific development of the models themselves to the exclusion of supporting the research, quantitative validation studies and infrastructure needed to improve the sophistication and scope of behaviors that can be represented in high quality models. As evidenced in this book and in the broader scientific community, considerable progress has been made in the development of integrated cognitive architectures. As also evidenced in this book and in the broader scientific community, considerable additional research is needed in order to achieve the desired levels of robustness, integrative fidelity, validity, parsimony, inspectability, interpretability, and cost effectiveness. These research directions are more than worthwhile. They are imperative.

5.6 References

Anderson, J. R. (1993). *Rules of the Mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. & Lebiere, C. L. (2003). The Newell test for a theory of mind. *Behavioral & Brain Science* 26, 587-637

Byrne, M. D., & Anderson, J. R., (1998). Perception and action. In J. R. Anderson and C. Lebiere (Eds) *Atomic Components of Thought* (pp. 167-200). Mahwah, NJ: Lawrence Erlbaum Associates.

Chong, R. S., & Laird, J. E., Identifying dual task executive process knowledge using EPIC-Soar. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 107-112). Mahwah, N.J.: Lawrence Erlbaum Associates.

Deutsch, S. E., Cramer, N. L., & Feehrer, C. E. (1994). *Research, development, training, and evaluation support. Operator model architecture (OMAR).* BBN Report 8019. Cambridge, MA: BBN Technologies.

Estes, W. K. (2002). Traps in the route to models of memory and decision. *Psychonomic Bulletin & Review, 9*(1), 3-25.

Gluck, K. A., Pew, R. W., (Eds.) (in press) *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation.* Mahwah, NJ: Lawrence Erlbaum Associates.

Gobet, F. & Ritter, F. E. (2002). Individual data analysis and unified theories of cognition: A methodological proposal. In N. Taatgen and Aasman, (Eds.) *Proceedings of the 3rd International Conference on Cognitive Modelling.* Veenendaal (NL) Universal Press, 150-157.

Hart, S., & Staveland, L. (1988). Development of the NASA-TLX: Results of empirical and theoretical research. In P. Hancock and N. Meshkati (Eds.), *Human mental workload.* Amsterdam: North-Holland, 139-184.

Hudlicka, E. (2002). "This time with feeling: integrated model of trait and state effects on cognition and behavior." *Applied Artificial Intelligence* 16 (7-8), 1-31.

Johnson, T. R. (1997). Control in ACT-R and Soar. In *Proceedings of the 19th Annual Meeting of the Cognitive Science Society* (pp. 343-348). Mahwah, NJ: Lawrence Erlbaum Associates.

Jones G. (1996). The architectures of Soar and ACT-R, and how they model human behaviour. *Artificial Intelligence and Simulation of Behaviour Quarterly, 96,* 41-44.

Kieras, D. E., & Meyer, D. E. (2000). The role of cognitive task analysis in the application of predictive models of human performance. In J. M. C. Schraagen, S. E. Chipman, & V. L. Shalin (Eds.), *Cognitive task analysis.* Mahwah, NJ: Lawrence Erlbaum Associates.

Lallement, Y. & John, B. E. (1998) Cognitive architecture and modeling idiom: An examination of three models of the Wicken's task. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society,* Mahwah, NJ: Lawrence Erlbaum Associates.

Lane, N., Strieb, M. I., Glenn, F. A., & Wherry, R. J. (1980) "The human operator simulator: an overview." *Proceedings of Conference on Manned Systems Design* (pp. 1-39). Freiburg.

Micro Analysis and Design (2004) http://www.maad.com/index.pl/products

Morrison, J. E. (2003). *A review of computer-based human behavior representations and their relation to military simulations (IDA Paper P-3845).* Alexandria, VA: Institute for Defense Analyses.

Pew, R. W. & Mavor, A. S. (Eds.) (1998). *Modeling human and organizational behavior: Applications to military simulations*. Washington, DC: National Academy Press.

Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R. M., Gobet, F., & Baxter, G. D. (2003). *Techniques for modeling human performance in synthetic environments: A supplementary review (HSIAC-SOAR-2003-01)*. Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center.

Vidulich, M.A. & Tsang, P.S. (1986). Collecting NASA Workload Ratings: A Paper-and-pencil Package (Version 2.1), Working Paper. Moffett Field, CA: NASA Ames Research Center.

Appendix A: Experimenter's Scripts and Demos

This section contains the scripts that the experimenters followed when administering the experiment to the Experimental and Control groups and the text for the AVI demos. Note that underlined items indicate actions the experimenter takes. Non-underlined items are passages that the experimenter reads aloud to the subject. Also note that "S" stands for subject and "E" stands for Experimenter.

A.1 Experimenter's Script for Experimental/Dual Group

AMBR3 Experimenter's Script
for EXPERIMENTAL/DUAL GROUP

Summary of Testing Steps

Consent Form

Background Form

Color Vision Test

Introduction

Training: Hand-off Task

Explanation

Automated Demo: 3HAND-OFF (flawless)

Review of Priorities
Automated Demo: 3HAND-OFF (with errors)
Coached Practice: 6HAND-OFF
Uncoached Practice: 19HAND-OFF

Training: Dual Task

Explanation
Automated Demo: 3HAND-OFF+3MAGENTA (get 2 right, 1 wrong, no delays)
Review of Priorities
Automated Demo: 3HAND-OFF+3MAGENTA
(1 no-response; 2 delays; 1 correct, 1 wrong)
Coached Practice: 5HAND-OFF + 5MAGENTA
Quiz – Dual Task

Break

Trials

[each has 12 "Hand-off" tasks and 16 "Altitudes" for medium workload condition;
16 "Hand-off" tasks and 16 "Altitudes" for high workload condition]

Trial 1: No
Trial 2
Trial 3
Trial 4: Workload Questionnaire, -Break-
Trial 5
Trial 6

Trial 7
Trial 8: Workload Questionnaire

Transfer Task (no demo)

Debrief

On-line
Oral (E take notes)

Payment Receipt Form

Non-Disclosure Request

Debriefing Handout

Prior to Subject's Arrival:

OBTAIN SUBJECT NUMBER (from Master List)

Turn on speakers (Sound should come from stereo speakers, not out of computer)

Put "Testing" notice on door.

Put Telephone on "Do Not Disturb".

However, if only one subject is being tested at that time, use the phone as a monitor into the Observation room by calling into that room, putting both phones on "speaker", and putting the Observation room phone on "Mic-off" (If there is a Baby Monitor into the Observation room, use that instead of the telephone.). [This sentence refers to the BBN set-up.]

COMPUTER SET-UP

--To close out previous session On Simulation Radar Frame: File-Close (from menu bar)

On Radar Model Client Frame: File-Exit Application Client (from menu bar)

On Allegro Common Lisp console: Close (Windows X, top right corner)

On Experiment window: Close (Windows X, top right corner)

--To start new session

Double Click on icon labeled AMBR Phase 3

Type in subject number

If you get a message saying the subject # has already been used, enter a different #

If you need to exit from program at this point, type any negative # (e.g. -3).

FORMS

--Consent Form
--Background Form (S fills it out)
--Color Vision Test
--Payment Receipt Form
--Debriefing Handout

TRAINING MATERIALS

--Penalty Points (Hand-off Task)

--Feedback Sheet (Smiley Face)

--Penalty Points (Dual Task)

--Features List for Transfer Test

REMUNERATION

--cookies and soda and --checks

BACK UP OF THE DATA --should be done each evening

TO REPLACE A SUBJECT

--Write down in note book the reason why subject is being replaced

--Go into Explore

--Go into the E (or other) drive

--Go into ambr-phase-3

--Click on "data", look in right site of window for the files.

--Find and delete all the files with the relevant subject number

(e.g. For subject 90, find: Exp3-sub-90-cat...)

[Note: there is a moment between clicking away the previous practice, and the Dual-demo Screen coming up.]

TRAINING: DUAL TASK

EXPLANATION

Point to on-line screen shot with colored and magenta aircraft and point to appropriate areas as you mention them:

Just to complicate things, sometimes an aircraft in your sector will turn magenta. Magenta means that the aircraft is requesting an altitude change. You will accept or reject the altitude change request by clicking on the Accept Altitude Request or Reject Altitude Request button (we'll talk about how you decide which one in a minute), and then you'll click on the Aircraft and the Send. This is another action, like the Welcome, that does not require clicking on the Air Traffic Control center.

Your goal in this experiment is to learn to respond correctly to requests for altitude changes.

You will have to figure out how to respond to altitude requests based on the feedback you receive and three properties of the aircraft. The three properties that you should pay attention to are Percent Fuel Remaining, which will always be either 40%, abbreviated to 40 **(point to screen)** or 20%, abbreviated to 20, we don't have an example of that on this screen – you'll see it on the demo – Size of the aircraft, large or small, abbreviated to L for large or S for small, like T-shirt sizes, and Turbulence, 1 for relatively low turbulence and 3 for moderate turbulence. The properties will be visible only while the plane is magenta, and for a few seconds after you respond to the altitude request.

Those are the only three properties you need to consider. This is not a real Air Traffic Control system. There are no other features that are in any way relevant. When the aircraft turns magenta, you should accept or reject the Altitude Change Request based on the properties of the aircraft. If you do not know the answer, take a guess. You'll have to guess on the first trial because you'll have no way of knowing what the correct answer is. You may see a pattern to the properties of the planes you should accept. It may be complex, or there may be no pattern at all. Even if you don't see a pattern, the planes that you are supposed to accept, as defined by the three properties, will be the same ones throughout the session, and will not vary from trial to trial.

Now, here is the feedback you get:

Show handout: Feedback (smiley face and X).
A smiley face next to the plane with a high tone (like a bell) means your response was correct. An X with a low tone (like a growl) means your response was wrong.

*****PRESS SPACE BAR TO CONTINUE*****

AUTOMATED DEMO: 3HAND-OFF + 3MAGENTA (NO Altitude DELAYS)

Now I'm going to start up a demonstration that will add the Altitude Change Request task to the Hand-off task you've already seen. Whatever you learn here regarding the magenta planes may be different in the actual trials.

Press "start" to see demo. ---run demo---

[Note: to know you are seeing the correct demo, the Demo dialogue starts with:
"In this demo, I'm going to add the Altitude Change Request task to the hand-off task you've already seen..."]

[Note: There is a pause before seeing the Penalty Screen – wait for the Penalty Screen to come up before closing the application.]

PRESS "CLOSE" TO CONTINUE

EXPLANATION OF PRIORITIES

Show Handout: Penalties (Dual Task) Let's go over what your priorities should be in doing the total task.

Your top priority should be to answer an altitude request in a timely manner. If you don't respond before the magenta plane returns to white, you will be penalized 200 points. So, when you see a magenta aircraft start flashing, you should attend to it immediately because it means the opportunity to respond to an altitude request is running out. To make matters worse, when the magenta plane is flashing, all the other buttons such as Accepting and Transferring are grayed out, yet all the planes will keep moving, so you will not be able to respond to them or attend to an aircraft that is about to turn red, until you respond to the altitude request.

If you don't know the correct answer, take a guess. You will get 100 points for an incorrect answer, but you will get 200 points for not responding at all, so it's wise to take a guess.

Your next highest priority is to keep an aircraft from going on hold. Each time an aircraft goes on hold, you are penalized 50 points.

Your third priority is to get a red aircraft out of hold. You get 10 points for each minute the aircraft is on hold.

Your lowest priority is welcoming an aircraft. There is no serious penalty associated with not welcoming an aircraft. It will not turn red. The penalty for not welcoming an aircraft is 1 point per minute.

Also keep in mind the few additional penalties. You get 10 points for sending the same message twice, 10 points for clicking on an Air Traffic Control center when it's not required by the

message template, and 10 points for sending an incorrect message (e.g. using the Contact ATC button when you should be Welcoming the plane.).

AUTOMATED DEMO: 3HAND-OFF+3MAGENTA (with DELAYED RESPONDING)

Now I'll do the demo again so you can see what happens if I take too long to respond to the altitude request. Again, whatever you learn here regarding the magenta planes may be different in the actual trials.

Press "start" to see demo. —run demo—

[Note: to know you are seeing the correct demo, the Demo dialogue starts with:
"This time, I'm going to demonstrate the Altitude Change Request task with the hand-off task again, but I'll delay in responding…"]

[Note: There is a pause before seeing the Penalty Screen – wait for the Penalty Screen to come up before closing the application.]

*****PRESS "CLOSE" TO CONTINUE*****

Coached Practice: 5HAND-OFF+5MAGENTA

Now I'll let you try out the full task. Here is a practice scenario that has both the Hand-off-task we first talked about and practiced, and the one with the magenta altitude-request planes. I'll coach you through this one. I'll make sure you get the Hand-off task correctly, and coach you through the correct sequence of altitude request mouse clicks, but I'll let you decide whether to accept or reject the altitude request.

[Have the subject start the practice when they're ready.]

When the practice is through: Do you understand the penalty sheet? Any questions?

QUIZ: DUAL TASK
[Note: Make sure you've collected all the handouts before giving the quiz.]

Now I'd like you to take this little quiz about the rules of the Air Traffic Controller task. This is to make sure you really understand what you need to do before you start. Don't worry if you don't know all the answers. You'll see the answers to any that you got wrong as soon as you finish answering all the questions.

There are twenty-three questions. Even though it will look like you are supposed to submit your answers after the first nine questions, just keep scrolling until you get to question 23.

Do you have any questions?

BREAK: 10 MINUTES

[Note: Subject may bring snacks into testing room, but Experimenter should request that the Subject doesn't partake of snacks or drinks while in the midst of a scenario. If the room allows, have the Subject leave the snack and/or drink on a different table from the one the experiment is being run at.]

[Please keep an eye on the time and keep it to the 10 minutes if possible.]

(If subject has a snack) "Please leave your snack over here." **(out of reach.)** "It's okay to get up between trials to have your snack, but we don't want you to be distracted during the trials."

TRIALS

Now we'll start the actual trials. This is a brand new problem. What was correct before in the demonstrations and practices for the altitude change requests may not be correct now. However, starting now, whatever you learn from the smiley faces will remain the same for the rest of today's session. Remember, the important features are Percent of Fuel Remaining, Size, and Turbulence. Nothing else is relevant. You may see a pattern to the properties of the planes you are supposed to accept. It may be complex, or there may be no pattern at all. Even if you don't see a pattern to the "Accepts", the planes you are supposed to accept, as defined by the three properties, will be the same throughout the session and will not vary from trial to trial.

There will be eight trials, and they will last approximately ten minutes each. Periodically after a trial, a Workload Questionnaire will pop up. The instructions are on it. Please answer all the questions on it each time.

ONE OTHER THING: YOU ARE NOT ALLOWED TO TAKE NOTES.

Any questions?

While we're doing the remaining eight trials you're on your own. I'll just be sitting here reading in case there are any problems with the system.

Trial 1: Workload Questionnaire
Trial 2
Trial 3
Trial 4: Workload Questionnaire, -Break-
Trial 5
Trial 6
Trial 7
Trial 8: Workload Questionnaire

Note: This is what subjects see, on-line, for the workload questionnaire:

Workload Questionnaire (1=very low, 7=very high)

Mental Demand: How mentally demanding was the task?
Physical Demand: How physically demanding was the task?
Temporal Demand: How hurried or rushed was the pace of the task?
Performance Errors: How likely were you to make mistakes on this task?
Effort: How hard did you have to work to accomplish your level of performance?
Frustration: How insecure, discouraged, irritated, and annoyed were you?

TRANSFER TASK ----(no demo)

[Note: It is okay to read these directions while the 'system is loading' after the previous task.]

Now we are going to change the task somewhat.

You'll be handling only altitude change requests. You won't have to take care of any other tasks like Transfers and Accepts. No other plane colors besides white and magenta will appear on the radar screen. Also, the planes will not be moving.

Here's what we'll be doing now:

--Give handout of 'Feature Values'--

For the Percent Fuel Remaining, the fuel remaining may be at 20% and 40% as before. Now we're adding 10%, 30% and 50%. They will be written under the plane icon as the others have been; so you'll see 10, 20, 30, 40, or 50.

For the Size of the planes, you've seen Small and Large, written S and L. We're adding Extra Small, Medium and Extra Large written XS, M, or XL (like T-shirt sizes). You can see these on the handout too.

You had only 1 and 3 for Turbulence before. Now, the Turbulence will be: 0, 1, 2, 3, and 4, with 4 most turbulent.

You will be doing the same as you've been doing, Accepting or Rejecting an Altitude Change Request after the plane turns magenta. You'll still click on the Accept/Reject Altitude buttons, the Aircraft, and Send the message.

However, now there will be no feedback – no smiley faces or Xs, and no high or low tones, and you will not have the time constraints – the planes will not blink to say time is running out, and they will stay magenta and not return to white until you respond to the altitude request.

What was right before is still right. With this task, we are asking you to make your best judgment as to whether you should Accept the altitude change request or Reject it, based on what you learned before, and then go on to the next request that comes along. We want to know how you would extend what you've learned to new cases that you haven't seen before, but that you might have some idea how to handle.

--- Start Transfer Scenario ---

DEBRIEF

ON-LINE

I'd like you to answer the questions on the computer screen as best you can.

[Read what the S is writing so you can follow-up on anything. Don't be afraid to look over their shoulders if you can't see the screen;
If they change any of their responses as they're typing, they may have begun an idea that would be worth pursuing on the oral debrief component. Make a note to yourself to ask them about it during the Oral Debrief.]

These are the open-ended questions the subject will be asked to fill out on-line on the <u>first screen</u>:

On the last of the eight trials (the ones with the moving planes and smiley faces):

- **Q1. How did you decide whether to accept or reject an altitude change request?**

- **Q2. Did your strategy change over the 8 trials [Please explain]**

On the very last task (the one with the stationary planes and the extra properties):

- **Q3. How did you decide whether to accept or reject an altitude change request?**

These are the questions the subject will be asked to fill out on the <u>second screen</u>:

Here are some additional, more specific questions:

On the last of the eight trials (the ones with the moving planes and smiley faces):

- **Q1. Did you use a rule? (check) Yes No**
  [Note: the subject will not be able to continue until checking either Yes or No.]

- **Q2 . If yes, I accepted an altitude request when:**

- **Q3 . If no, what did you do?**

DEBRIEF – continued

When the subject is done with the on-line debrief, and has clicked 'ok', RETRIEVE THE MOUSE so they don't click the 'Please Wait' button and cause you to lose sight of the answers they just typed

You will now see the subject's answers to the previous on-line questions.

[Note: you will have to scroll vertically and horizontally to view the responses.]

***In addition*, you will see a 1, 3 or 6 to let you know what category structure the subject had. Below that you will see the 4 correct Accepts and the 4 correct Rejects that the subject should have made during each of the 8 trials (s)he was given. Do not call the subject's attention to this information. It's there so that you, the experimenter, can better follow what the subject is saying.**

They will appear on the screen like this (example):

6

4000 L 1 A
4000 L 3 A
4000 S 3 A
4000 S 1 A
2000 L 1 R
2000 L 3 R
2000 S 1 R
2000 S 3 R
—
[The first set are the Accepts (A), the second, the Rejects (R).]

ORAL (E take notes capturing as much of what S says as possible)

Follow up on any responses from the On-Line form that aren't clear or need further exploration or can elicit further insights on what strategy(ies) the S was using.

PAYMENT

That's all we'll be doing. Thank you for your help today.

[give check, Payment Receipt Form, and Debriefing Handout]

NON-DISCLOSURE REQUEST

There is just one thing we'd like to request. Please do not talk about this study to any one who might potentially be a participant. We would like all participants to be new to the task. So, we would appreciate it if you did not discuss any of the details. OK?

If subjects ask what the study is about, say we are not at liberty to discuss it right now, but would be happy to send them the report when it is ready.

(Get their email or home address if they want a copy)

Ask the subject: Is it okay for us to contact you if we have questions later?

[Reminder: Back-up the data when the subject leaves.]

Questionnaires/Handouts
(Experimental Group)

[on Cambridge Focus letterhead] **#:________**

CONSENT FORM

BBN Technologies ("BBN") is conducting an experiment to analyze how people learn in complex situations. Participants will be presented with a sequence of events and asked to devise solutions to those situations. The experimenter will ask questions related to the situations and the participant will be expected to answers questions periodically during the session. BBN may record and/or videotape each session and use it for both data analysis and documentation of the testing procedure.

You, as a participant in the BBN experiment, understand that you will participate in a session, for up to three and one-half hours in length.

You understand you will be asked to interact with a sequence of events displayed on a computer screen and answer questions regarding the solutions you devise in response to those events.

You understand you may be recorded and/or videotaped as you participate in the study.

You understand that all materials and/or information used and disclosed is or will become the property of BBN and is proprietary and confidential information. You agree not to disclose such proprietary and confidential information to any third party. In addition you understand, BBN is and shall be the exclusive owner of any and all right, title, and interest (including copyright) in and to any materials developed by BBN, which incorporates any of the information and/or materials resulting from your participation in such a session.

You understand you may change your mind about taking part in this study at any time during the session. If this is the case and you decide to stop, you will promptly inform BBN's experimenter. You understand that if you elect to stop you will only be paid an amount commensurate with the amount of time actually spent participating. (Due to the nature of the study once the session begins you may not participate a second time or make up any part of a session at a later date.)

You do not expect to receive, and have not been promised, compensation beyond the amount agreed upon in this study.

You hereby certify you are at least eighteen years old, you are a U.S. citizen, and you have no physical or mental conditions that would hinder your participation in this study.

By your signature below you hereby agree you have read and understand the terms and conditions stated herein and that you want to take part in this study. You also understand that by your signature below that you are agreeing it is your responsibility to ensure you are in compliance with the stated terms and conditions set forth in this Consent Form.

______________________________ ______________________________

Participant Name (please print) Participant Signature/Date

Name:______________________________ **Participant**
#:________

BACKGROUND FORM

Date:______________ **Time:**______

Sex: M F

Age: ____________________

Education:

Major: __________________________________

Class Standing: Freshman______ Sophomore______ Junior______ Senior______

Other (please explain)____________________________________

GPA: ___________ **SAT Score:** ___________ **ACT Score:** ___________

Do you own a personal computer? YES:______ NO:______

Have you ever played any air traffic control radar games? Yes:______ No:______

Have you had any experience(s) which have made you familiar with air traffic control equipment, and/or terminology? Please explain:

____

____

#:________

COLOR VISION TEST

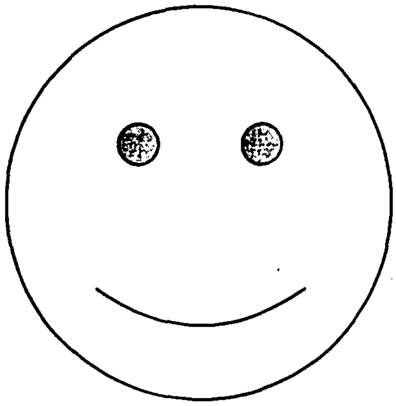
Please locate each of the following aircraft on the screen and write down its color in the space provided.

Aircraft	**Color**
1. AFR940	____________
2. OLY492	____________
3. WNG736	____________
4. NWA190	____________
5. LUF450	____________
6. USA475	____________
7. OLY313	____________

PENALTY POINTS (Hand-off Task)

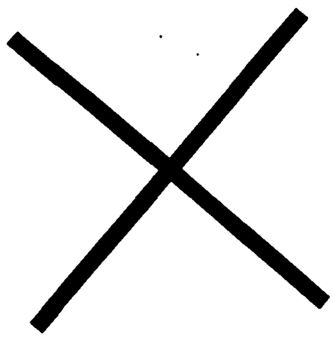
1. Prevent AC from holding either while incoming or outgoing
  50 points each time an AC turns red
2: Get AC out of holding
  10 points for each minute AC stays red
3: Welcome an Aircraft
  1 point for each minute aircraft not welcomed
4: Additional Penalties
  10 points for duplication of a message
  10 points for clicking on an Air Traffic Control center when it's not required
  10 points for sending an incorrect message

FEEDBACK



with HIGH tone
("bell")

means CORRECT



with LOW tone
("growl")

means INCORRECT

PENALTY POINTS (Dual Task)

1. Answer altitude request in timely fashion
200 points each time no response in allotted time
2. Answer altitude request correctly
100 points for each incorrect answer
3. Prevent AC from holding either while incoming or outgoing
50 points each time an AC turns red
4: Get AC out of holding
10 points for each minute AC stays red
5: Welcome an Aircraft
1 point for each minute aircraft not welcomed
6: Additional Penalties
10 points for duplication of a message
10 points for clicking on an Air Traffic Control center when it's not required
10 points for sending an incorrect message

FEATURE VALUES

PERCENT FUEL REMAINING:

% Fuel Remaining:	10%	20%	30%	40%	50%
Written as	10	20	30	40	50

SIZE of Plane:

Size:	Extra Small	Small	Medium	Large	Extra Large
Written as	XS	S	M	L	XL

TURBULENCE:

Least turbulent	0	1	2	3	4	Most turbulent

#:________

AMBR Payment Receipt

This is to certify that I have received the dollar amount that I was promised and agree that the payment is in full consideration for my participation.

All materials from this session are solely for the use of BBN Technologies and its research partners.

I agree to keep the nature and content of this study session confidential.

Name (Please Print): ________________________________

Signature: ____

Date: __________________________

*****_Thank you so much for participating in our study._*****

Would you like to receive an abbreviated copy of the results when the experiment is complete? (circle one)

YES NO

If yes, please give us your email or home address:

Debriefing Handout

Thank you for participating in today's experiment. You have taken part in a study where participants are asked to make decisions in the context of an air traffic control-like simulation. The task featured in this study was designed to collect data on how humans make decisions, and it is not a true air traffic control task. Based on the data collected with this testbed, we will attempt to build and evaluate computer simulations (or models) of people making decisions. To achieve this goal, your data will be combined with other participants' data and averages will be provided to software modelers. Please note that we are using these data to evaluate the computer simulations that are built. ***We are not evaluating you or your performance.***

If you are interested in more information about this project, we will be happy to provide you with an abbreviated abstract of the results once the data collection is complete. Let us know before you leave if you would like to receive an abstract.

Finally, we would like to ask you not to discuss the details of this experiment with anyone else. It won't be a fair test for the computer simulations if some of the people know details about the task before they begin the experiment.

Thank you for your cooperation and your time!

AMBR3 – Subject Numbers for Experimenters

BBN

Subject Number	Condition		Subject Number	Condition
1	E		82	E
2	E		83	E
3	C		84	C
4	E		85	E
5	E		86	E
6	C		87	C
7	E		88	E
8	E		89	E
9	C		90	C
10	E			
11	E			
12	C			
13	E			
14	E			
15	C			
16	E			
17	E			
18	C			
19	E			
20	E			
21	C			
22	E			
23	E			
24	C			
25	E			
26	E			
27	C			
28	E			
29	E			
30	C			
31	E			
32	E			
33	C			
34	E			
35	E			
36	C			

AMBR3 – Subject Numbers for Experimenters

UCF

Subject Number	Condition		Subject Number	Condition
37	E		73	E
38	E		74	E
39	C		75	C
40	E		76	E
41	E		77	E
42	C		78	C
43	E		79	E
44	E		80	E
45	C		81	C
46	E			
47	E			
48	C			
49	E			
50	E			
51	C			
52	E			
53	E			
54	C			
55	E			
56	E			
57	C			
58	E			
59	E			
60	C			
61	E			
62	E			
63	C			
64	E			
65	E			
66	C			
67	E			
68	E			
69	C			
70	E			
71	E			
72	C			

A2. Experimenter's Script for Control/Single Task Group

AMBR3 Experimenter's Script

For CONTROL GROUP

Summary of Testing Steps

Consent Form
Background Form
Color Vision Test
Introduction

Training
Explanation
Automated Demo: 3WHITE + 3MAGENTA (2 right, 1 wrong, no delays)
Review of Priorities
Automated Demo: 3WHITE + 3MAGENTA (1 no-response; 2 delays; 1 correct, 1 wrong)
Coached Practice: 5 WHITE + 5 MAGENTA
Quiz: Concept Task

Break

Trials (90 min)
[each has 16 Altitudes Requests]
Trial 1: Workload Questionnaire
Trail 2
Trial 3
Trial 4: Workload Questionnaire -Break-
Trial 5
Trial 6
Trial 7
Trial 8: Workload Questionnaire

Transfer Task (no demo)
Debrief

On-line
Oral (E take notes)

Payment Receipt Form
Non-Disclosure Request
Debriefing Handout

Prior to Subject's Arrival

OBTAIN SUBJECT NUMBER (from Master List)

Turn on speakers (Sound should come from stereo speakers, not out of computer)

Put "Testing" notice on door.

Put Telephone on "Do Not Disturb".

However, if only one subject is being tested at that time, use the phone as a monitor into the Observation Room by calling into that room, putting both phones on "speaker", and putting the Observation Room phone on "Mic-off" (If there is a Baby Monitor into the Observation room, use that instead of the telephone.). [This refers to the BBN set-up.]

COMPUTER SET-UP

--To close out previous session

On Simulation Radar Frame: File-Close (from menu bar)

On Radar Model Client Frame: File-Exit Application Client (from menu bar)

On Allegro Common Lisp console: Close (Windows X, top right)

On Experiment window: Close (Windows X, top right)

--To start new session

Double Click on icon labeled AMBR Phase 3

Type in subject number

If you get a message saying the subject # has already been used, enter a different #

If you need to exit from program at this point, type any negative # (e.g. -3).

FORMS
--Consent Form
--Background Form (S fills it out.)
--Color Vision Test

--Payment Receipt Form

--Debriefing Handout

TRAINING MATERIALS

--Feedback Sheet (Smiley Face)

--Features List for Transfer Test

REMUNERATION --cookies and soda --checks

BACK UP OF THE DATA

--should be done each evening

TO REPLACE A SUBJECT

--Write down in note book the reason why subject is being replaced

--Go into Explore

--Go into the E (or other) drive

--Go into ambr-phase-3

--Click on "data", look in right site of window for the files.

--Find and delete all the files with the relevant subject number

(e.g. For subject 90, find: Exp3-sub-90-cat...)

CONSENT FORM
BACKGROUND FORM

[move subject's keyboard out of the way]
(Hand out 2 forms for participant to fill out)

COLOR VISION TEST

I'll be reading this script to you to make sure that I don't leave out any information.

Point to on-line screen shot with colored and magenta aircraft.
******IF YOU ACTIVATE ANY OF THE BUTTONS, THE TASK WILL 'HANG'******
[if the task does hang, click inside the radar screen where the planes are, and that should allow you to continue]
These are some of the colors you will be seeing today. To make sure that you have no trouble telling them apart, I'd like you to take this simple color naming test.

(Hand out test form)
Please locate each of the following aircraft on the screen and write down its color in the space provided.

(Score test: If participant has difficulty on this test, ask if participant is aware of having any problems with colors or color-blindness and note the response. If participant is willing, continue with the session anyway. After the practice trials, ask the participant if the colors are a problem. If participant reports any difficulty, give him/her the choice of continuing or stopping. If participant wishes to continue, do the runs and report the results to BBN. After all the data are in, BBN may set a criterion on the color test and replace participants who did not meet it.)

COLOR VISION TEST ANSWER KEY

Aircraft	**Color**
1. AFR940	brown/orange/olive/khaki/Army green/golden brown/tan
2. OLY492	yellow
3. WNG736	white
4. NWA190	red
5. LUF450	blue/teal/turquoise/greenish-blue [if say greenish-blue, make certain they are seeing this color distinct from #'s 1 and 7]
6. USA475	magenta/pink/purple/lilac
7. OLY313	green/lime green/neon green

*****PRESS SPACE BAR TO CONTINUE*****

INTRODUCTION

In this experiment we are collecting data on human performance that will be used by researchers to build realistic human performance computer models. We are looking at performance under a variety of conditions that make learning more or less difficult. We are interested in how people in general perform on these tasks and not in your individual performance.

TRAINING

EXPLANATION

The task you will be doing is a modified form of an Air Traffic Controller's task. As you will see, it captures some parts of the real job of an air Traffic Controller, but has been greatly modified for the experiment. We are not actually studying air traffic control. This is a learning study, and we are using air traffic control as the context.

[Make sure 'training' screen-shot is up.]

Describe Concept Task. Point to on-line screen shot with white and magenta aircraft:

You are the aircraft controller in this central sector, bounded by the yellow line **(point).** There are controllers in the 4 adjoining sectors, N, S, E, and W **(point).** Aircraft will be entering your sector, flying around in your sector, and leaving your sector **(point).** In this simplified task, you will not be concerned with collisions.

Sometimes an aircraft in your sector will turn magenta. Magenta means that the aircraft is requesting an altitude change. You will accept or reject the altitude change request by clicking on the Accept Altitude Request or Reject Altitude Request buttons and then you'll click on the Aircraft and the Send. You will see the Send button active in the demo.

Now how will you decide whether to accept or reject the altitude requests? Your goal in this experiment is to learn to respond correctly to requests for altitude changes.

You will have to figure out how to respond to altitude requests based on the feedback you receive and three properties of the aircraft. The three properties that you should pay attention to are Percent Fuel Remaining, which will always be either 40% **(point to screen)** or 20%, we don't have an example of the 20 on this screen – you'll see it on the demo, Size of the aircraft, large or small, abbreviated to L for large or S for small, like T-shirt sizes, and Turbulence, 1 for relatively low turbulence and 3 for moderate turbulence. The properties will be visible only while the plane is magenta, and for a few seconds after you respond to the altitude request.

Those are the only three properties you need to consider. This is not a real Air Traffic Control system. There are no other features that are in any way relevant. When the aircraft turns magenta, you should accept or reject the Altitude Change Request based on the properties of the aircraft. If you do not know the answer, take a guess. You'll have to guess on the first trial because you'll have no way of knowing what the correct answer is. You may see a pattern to the properties of the planes you should accept. It may be complex, or there may be no pattern at all. Even if you don't see a pattern, the planes that you are supposed to accept, as defined by the three properties, will be the same ones throughout the session, and will not vary from trial to trial.

Now, here is the feedback you get: **Show handout: Feedback (smiley face and X)**
A smiley face next to the plane with a high tone (like a bell) means your response was correct. An X with a low tone (like a growl) means your response was wrong.

*****PRESS SPACE BAR TO CONTINUE*****

[Make sure the speakers are on.]

AUTOMATED DEMO: 3WHITE + 3 MAGENTA (NO DELAYS)

Now I'm going to start up a demonstration that will show how to do the task. Whatever you learn here about which altitude requests to accept may be different in the actual trials.

[Make sure your own mouse pointer is pushed out of the way.]

Press "start" to see demo. —run demo —

[Note: to know you are seeing the correct demo, the Demo dialogue starts with:
"In this demonstration, I will show how to handle the Altitude Change Request task".]

[Note: There is a pause before seeing the Penalty Screen – wait for the Penalty Screen to come up before closing the application.]

*****PRESS "CLOSE" TO CONTINUE*****

EXPLANATION OF PRIORITIES [There is no handout.]

Let's review what your priorities should be in doing the task.

Your top priority should be to answer an altitude request in a timely manner. If you don't respond before the magenta plane returns to white, you will be penalized 200 points. So, when you see a magenta aircraft start flashing, you should attend to it immediately because it means the opportunity to respond to an altitude request is running out.

If you don't know the correct answer, take a guess. You will get 100 points for an incorrect answer, but you will get 200 points for not responding at all, so it's wise to take a guess.

You will also get a small number of penalty points if you try to respond or change your answer after the aircraft has turned back to white.

AUTOMATED DEMO: 3WHITE + 3MAGENTA DEMO (with NO RESPONSE + DELAYS)

Now you'll see the demonstration again, but this time it will show you what happens if you take too long to respond to the altitude requests.

Again, whatever you learn here regarding the magenta planes may be different in the actual trials.

Press "start" to see demo. —run demo—

[Note: to know you are seeing the correct demo, the Demo dialogue starts with:
"In this demonstration, I will show you what happens when I delay in responding to an Altitude Change Request or do not respond at all."]

[Note: There is a pause before seeing the Penalty Screen – wait for the Penalty Screen to come up before closing the application.]

*****PRESS "CLOSE" TO CONTINUE*****

COACHED PRACTICE
Now I'll let you try out the task. I'll coach you through this one and make sure you have the correct sequence of mouse clicks, but I'll let you decide whether to accept or reject an altitude request.

[Let the Subject start the practice when they're ready.]

QUIZ: ALTITUDE TASK
[Note: Make sure you've collected all the handouts before giving the quiz.]

Now I'd like you to take this little quiz about the rules of the Air Traffic Controller task. This is to make sure you really understand what you need to do before you start. Don't worry if you don't know all the answers. You'll see the answers to any that you got wrong as soon as you finish answering all the questions.

There are eight questions. Even though it will look like you are supposed to submit your answers after the first three questions, just keep scrolling until you get to question 8.

Do you have any questions?

BREAK: 10 MINUTES

[Note: Subject may bring snacks into testing room, but Experimenter should request that the Subject doesn't partake of snacks or drinks while in the midst of a scenario. If the room allows, have the Subject leave the snack and/or drink on a different table from the one the experiment is being run at.]

[Please keep an eye on the time and keep it to the 10 minutes if possible.]

(If subject has a snack) "Please leave your snack over here." **(out of reach.)** "It's okay to get up between trials to have your snack, but we don't want you to be distracted during the trials."

TRIALS

Now we'll start the actual trials. This is a brand new problem. What was correct before in the demonstrations and practices for the altitude change requests may not be correct now. However, starting now, whatever you learn from the smiley faces will remain the same for the rest of today's session. Remember, the important features are Percent of Fuel Remaining, Size, and Turbulence. Nothing else is relevant. You may see a pattern to the properties of the planes you are supposed to accept. It may be complex, or there may be no pattern at all. Even if you don't see a pattern to the "Accepts," the planes you are supposed to accept, as defined by the three properties, will be the same throughout the session and will not vary from trial to trial.

There will be eight trials, and they will last approximately ten minutes each. Periodically after a trial, a Workload Questionnaire will pop up. The instructions are on it. Please answer all the questions on it each time.

ONE OTHER THING: YOU ARE NOT ALLOWED TO TAKE NOTES.

Any questions?

While we're doing the remaining eight trials you're on your own. I'll just be sitting here reading in case there are any problems with the system.

RUN REAL TRIALS

Trial 1: Workload Questionnaire
Trial 2
Trial 3
Trial 4: Workload Questionnaire, -Break-
Trial 5
Trial 6
Trial 7
Trial 8: Workload Questionnaire

Note: This is what subjects see, on-line, for the workload questionnaire:

Workload Questionnaire (1=very low, 7=very high)

Mental Demand: How mentally demanding was the task?
Physical Demand: How physically demanding was the task?
Temporal Demand: How hurried or rushed was the pace of the task?
Performance Errors: How likely were you to make mistakes on this task?
Effort: How hard did you have to work to accomplish your level of performance?
Frustration: How insecure, discouraged, irritated, and annoyed were you?

TRANSFER TASK ----(no demo)

[Note: It is okay to read these directions while the 'system is loading' after the previous task.]

Now we are going to change the task somewhat.

You'll still be handling altitude change requests, but each of the properties of the plane will be expanded. Also, the planes will not be moving.

Here's what we'll be doing now:

Show handout: Features Values
For the Percent Fuel Remaining, the fuel remaining may be at 20% and 40% as before. Now we're adding 10%, 30% and 50%. They will be written under the plane icon as the others have been; so you'll see 10, 20, 30, 40, or 50.

For the size of the planes, you've seen Small and Large, written S and L. We're adding Extra Small, Medium and Extra Large written XS, M, or XL (like T-shirt sizes). You can see these on the handout too.

You had only 1 and 3 for Turbulence before. Now, the Turbulence will be: 0, 1, 2, 3, and 4, with 4 most turbulent.

You will be doing the same as you've been doing, Accepting or Rejecting an Altitude Change Request after the plane turns magenta. You'll still click on the Accept/Reject Altitude button, the Aircraft, and Send the message.

However, now there will be no feedback – no smiley faces or Xs, and no high or low tones, and you will not have the time constraints – the planes will not blink to say time is running out, and they will stay magenta and not return to white until you respond to the altitude request.

What was right before is still right. With this task, we are asking you to make your best judgment as to whether you should Accept the Altitude Change Request or Reject it, based on what you learned before, and then go on to the next request that comes along. We want to know how you would extend what you've learned to new cases that you haven't seen before but that you might have some idea how to handle.

— Start Transfer Scenario —

DEBRIEF

ON-LINE

I'd like you to answer the questions on the computer screen as best you can.

[Read what the S is writing so you can follow-up on anything. Don't be afraid to look over their shoulders if you can't see the screen;
If they change any of their responses as they're typing, they may have begun an idea that would be worth pursuing on the oral debrief component. Make a note to yourself to ask them about it during the Oral Debrief.]

These are the open-ended questions the subject will be asked to fill out on-line on the <u>first screen</u>:

On the last of the eight trials (the ones with the moving planes and smiley faces):

- **Q1. How did you decide whether to accept or reject an altitude change request?**
- **Q2. Did your strategy change over the 8 trials [Please explain]**

On the very last task (the one with the stationary planes and the extra properties):

- **Q3. How did you decide whether to accept or reject an altitude change request?**

These are the questions the subject will be asked to fill out on the <u>second screen</u>:

Here are some additional, more specific questions:

On the last of the eight trials (the ones with the moving planes and smiley faces):

- **Q1. Did you use a rule? (check) Yes No**
  [Note: the subject will not be able to continue until checking either Yes or No.]
- **Q2. If yes, I accepted an altitude request when:**
- **Q3. If no, what did you do?**

DEBRIEF -- continued

When the subject is done with the on-line debrief, and has clicked 'ok', RETRIEVE THE MOUSE so they don't click the 'Please Wait' button and cause you to lose sight of the answers they just typed

You will now see the subject's answers to the previous on-line questions.

[Note: you will have to scroll vertically and horizontally to view the responses.]

***In addition*, you will see a 1, 3 or 6 to let you know what category structure the subject had. Below that you will see the 4 correct Accepts and the 4 correct Rejects that the subject should have made during each of the 8 trials (s)he was given. Do not call the subject's attention to this information. It's there so that you, the experimenter, can better follow what the subject is saying.**

They will appear on the screen like this (example):

6

4000 L 1 A
4000 L 3 A
4000 S 3 A
4000 S 1 A
2000 L 1 R
2000 L 3 R
2000 S 1 R
2000 S 3 R
—
[The first set are the Accepts (A), the second, the Rejects (R).]

ORAL (E take notes capturing as much of what S says as possible)

Follow up on any responses from the On-Line form that aren't clear or need further exploration or can elicit further insights on what strategy(ies) the S was using.

PAYMENT

That's all we'll be doing. Thank you for your help today.

[give check, Payment Receipt Form, and Debrief Handout]

NON-DISCLOSURE REQUEST

There is just one thing we'd like to request. Please do not talk about this study to any one who might potentially be a participant. We would like all participants to be new to the task. So, we would appreciate it if you did not discuss any of the details. OK?

If subjects ask what the study is about, say we are not at liberty to discuss it right now, but would be happy to send them the report when it is ready.

(Get their email or home address if they want a copy)

Ask the subject: Is it okay for us to contact you if we have questions later?

[Reminder: Back-up the data when the subject leaves.]

Questionnaires/Handouts
(Control Group)

[on Cambridge Focus letterhead] #:________

CONSENT FORM

BBN Technologies ("BBN") is conducting an experiment to analyze how people learn in complex situations. Participants will be presented with a sequence of events and asked to devise solutions to those situations. The experimenter will ask questions related to the situations and the participant will be expected to answers questions periodically during the session. BBN may record and/or videotape each session and use it for both data analysis and documentation of the testing procedure.

You, as a participant in the BBN experiment, understand that you will participate in a session, for up to three and one-half hours in length.

You understand you will be asked to interact with a sequence of events displayed on a computer screen and answer questions regarding the solutions you devise in response to those events.

You understand you may be recorded and/or videotaped as you participate in the study.

You understand that all materials and/or information used and disclosed is or will become the property of BBN and is **proprietary and confidential information**. You agree not to disclose such proprietary and confidential information to any third party. In addition you understand, BBN is and shall be the exclusive owner of any and all right, title, and interest (including copyright) in and to any materials developed by BBN, which incorporates any of the information and/or materials resulting from your participation in such a session.

You understand you may change your mind about taking part in this study at any time during the session. If this is the case and you decide to stop, you will promptly inform BBN's experimenter. You understand that if you elect to stop you will only be paid an amount commensurate with the amount of time actually spent participating. (Due to the nature of the study once the session begins you may not participate a second time or make up any part of a session at a later date.)

You do not expect to receive, and have not been promised, compensation beyond the amount agreed upon in this study.

You hereby certify you are at least eighteen years old, you are a U.S. citizen, and you have no physical or mental conditions that would hinder your participation in this study.

By your signature below you hereby agree you have read and understand the terms and conditions stated herein and that you want to take part in this study. You also understand that by your signature below that you are agreeing it is your responsibility to ensure you are in compliance with the stated terms and conditions set forth in this Consent Form.

________________________ ________________________

Participant Name (please print) Participant Signature/Date

Participant #:________

BACKGROUND FORM

Date:______________ Time:______

Sex: M F

Age: ____________________

Education:

Major: ___________________________________

Class Standing: Freshman_____ Sophomore_____ Junior_____ Senior_____

Other (please explain)____________________________________

GPA: ___________ SAT Score: ___________ ACT Score: ___________

Do you own a personal computer? YES:_____ NO:______

Have you ever played any air traffic control radar games? Yes:_____ No:______

Have you had any experience(s) which have made you familiar with air traffic control equipment, and/or terminology? Please explain:

____

____

____

____

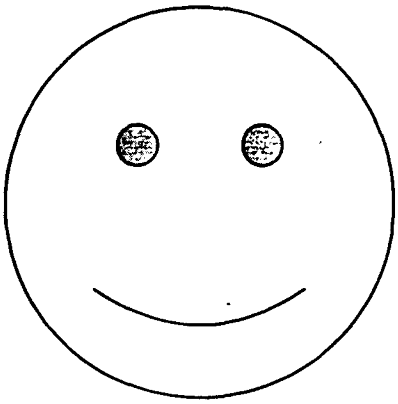
#_____

COLOR VISION TEST

Please locate each of the following aircraft on the screen and write down its color in the space provided.

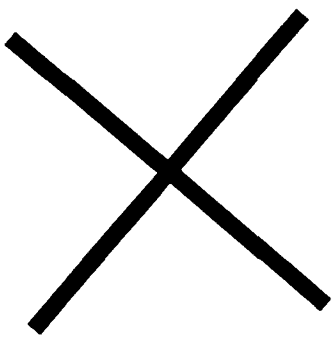
Aircraft	Color
1. AFR940	_______________
2. OLY492	_______________
3. WNG736	_______________
4. NWA190	_______________
5. LUF450	_______________
6. USA475	_______________
7. OLY313	_______________

FEEDBACK



with HIGH tone
("bell")

means CORRECT



with LOW tone
("growl")

means INCORRECT

FEATURE VALUES

PERCENT FUEL REMAINING:

% Fuel Remaining	10%	20%	30%	40%	50%
Written as	10	20	30	40	50

SIZE of Plane:

Size:	Extra Small	Small	Medium	Large	Extra Large
Written as	XS	S	M	L	XL

TURBULENCE:

Least turbulent	0	1	2	3	4	Most turbulent

#:________

AMBR Payment Receipt

This is to certify that I have received the dollar amount that I was promised and agree that the payment is in full consideration for my participation.

All materials from this session are solely for the use of BBN Technologies and its research partners.

I agree to keep the nature and content of this study session confidential.

Name (Please Print): ________________________________

Signature: ________________________________

Date: ____________________

*****_Thank you so much for participating in our study._*****

Would you like to receive an abbreviated copy of the results when the experiment is complete? (circle one)

YES   NO

If yes, please give us your email or home address

Debriefing Handout

Thank you for participating in today's experiment. You have taken part in a study where participants are asked to make decisions in the context of an air traffic control-like simulation. The task featured in this study was designed to collect data on how humans make decisions, and it is not a true air traffic control task. Based on the data collected with this testbed, we will attempt to build and evaluate computer simulations (or models) of people making decisions. To achieve this goal, your data will be combined with other participants' data and averages will be provided to software modelers. Please note that we are using these data to evaluate the computer simulations that are built. ***We are not evaluating you or your performance.***

If you are interested in more information about this project, we will be happy to provide you with an abbreviated abstract of the results once the data collection is complete. Let us know before you leave if you would like to receive an abstract.

Finally, we would like to ask you not to discuss the details of this experiment with anyone else. It won't be a fair test for the computer simulations if some of the people know details about the task before they begin the experiment.

Thank you for your cooperation and your time!

AMBR3 – Subject Numbers for Experimenters

BBN

Subject Number	Condition		Subject Number	Condition
1	E		82	E
2	E		83	E
3	C		84	C
4	E		85	E
5	E		86	E
6	C		87	C
7	E		88	E
8	E		89	E
9	C		90	C
10	E			
11	E			
12	C			
13	E			
14	E			
15	C			
16	E			
17	E			
18	C			
19	E			
20	E			
21	C			
22	E			
23	E			
24	C			
25	E			
26	E			
27	C			
28	E			
29	E			
30	C			
31	E			
32	E			
33	C			
34	E			
35	E			
36	C			

AMBR3 – Subject Numbers for Experimenters

UCF

Subject Number	Condition		Subject Number	Condition
37	E		73	E
38	E		74	E
39	C		75	C
40	E		76	E
41	E		77	E
42	C		78	C
43	E		79	E
44	E		80	E
45	C		81	C
46	E			
47	E			
48	C			
49	E			
50	E			
51	C			
52	E			
53	E			
54	C			
55	E			
56	E			
57	C			
58	E			
59	E			
60	C			
61	E			
62	E			
63	C			
64	E			
65	E			
66	C			
67	E			
68	E			
69	C			
70	E			
71	E			
72	C			

A.3 AVI Demo Scripts

["Secondary/Hand-off Task Demo 1 no delays"]
[for DUAL/Experimental] – COLOR ONLY
3-plane Color-demo 1—all correct

On this radar screen, you can see there are 3 aircraft, 2 are leaving my sector and 1 is entering my sector

[Incoming]
This one has just turned green. It is an incoming plane and it's asking us if we will accept it in, so I click the Accept Action Button, the airplane, the air traffic controller from the direction it's coming, and I Send the message.

[Outgoing]
Now we see that this plane is nearing the green line and has just now turned brown. That plane requires a transfer to the adjoining Eastern sector. So we click the Transfer button, the East Air Traffic Controller, the airplane, and then Send that message; and as you can see, right before we send it, in this template, it gives you the boxes that need to be filled in in case you forget.

[Incoming]
Over here, this plane has now turned blue and is saying Hello, as you can see here, so we are going to say we will Welcome you. We click the Action Button for Welcome, we click on the airplane only, not an Air Traffic Controller, and we send the message.

[Outgoing]
This one now has been accepted by the East, and we tell this airplane now to contact the East, we click the Contact, the Air Traffic Controller, the airplane, and send the message.

[Outgoing]
This one now is brown, which means it's ready to be transferred to the northern sector, so we click the airplane, the Northern sector, and send the message. As you've seen, I can click the plane and the Air Traffic Controller in either order, but you must always start with the Action Button, and of course, end with the Send button.

This plane as you see now is continuing into the sector, so all my actions were correct, and it did not turn red.

This one here is going to be heading out of the sector, so all of my actions were correct and it did not turn red.

This one here now up here [DAL121], has turned yellow, which means we must tell DAL to contact

["Secondary/Hand-off Task Demo 2 with delays"]
[for DUAL/Experimental] – COLOR ONLY
3-plane Color-demo 2—with errors

In this demonstration, I will show you what happens when you ignore some of the actions you need to take in the hand-off task
I'm going to deliberately delay things and make some errors so that you can see what happens to the planes when I do this, and what the scoring looks like.

So, again you see that this plane is heading out, this one is coming in;

[Accept 1st Green]
This one turned green so I'm going to use the Action Button called Accept to accept the plane into my sector, which bring up this template with the two boxes, and it means I have to put an airplane in one of them and an Air Traffic Controller in the other; and then I send the message.

[Let 1st Outgoing plane turn red]
This one has turned brown, but I'm going to let this plane go on past so that you'll see what happens when I don't handle the Transfer on this one.

[Skip Contact for 2nd Outgoing plane]
This plane that I accepted in has now asked us to Welcome him; now you can notice all these actions in this text area – this is an area that is showing what they are requesting and what we are doing as a response to that request, and you can see that this one is saying Hello to us.

So I'll also let him continue in so that you can see that a blue plane will not turn red and go in hold. A plane will not turn red if it's in the blue condition.

This plane has now asked for a transfer to the Northern sector, so I will hit the Transfer button, the air traffic controller, the airplane, and send this message along. But, I'm now going to skip the Contact with him so you'll see what happens.

Now, the brown plane is about to reach the yellow external boundary of my area and you'll see since I've not transferred it, it is going to turn red.

And the blue one here you can see, came in with no trouble at all.

This outgoing plane has now turned completely red, and it will just stay there until I begin the 2 part process that will Transfer him out of my sector and get him out of hold. First, I have to start a transfer of this plane to the Eastern sector, and send the message. So, what he is waiting for now, is for the Eastern sector to get back to me to tell me that they will accept him.

Over here, he is about to reach the yellow border of my area, and you'll see what happens to him if I have not performed the Contact on him.

Over here now, East has just accepted this airplane, but you'll see his nose is yellow, the plane now is red and yellow, so it tells me that I must perform the 2nd part of the Transferring process,

the Contact, to get him out of hold, so I click the Contact, the Air Traffic Controller, the airplane and send it.

And this guy also now [N] – you saw this one [E] turn pure red before because I had not even done a Transfer, this one is red and yellow because I did do the 1st part, the Transfer, now he's waiting for me to do the 2nd part, the Contact, and the need to do a Contact is indicated by the yellow color, so I click that, and I click the Air Traffic Controller, and send him on his way..

Now, in the mean time I can Welcome this plane in since all these other planes have already been handled, and I send this message. And you recall it's the Welcome Action Button, just the airplane, you can see up in the template, there is only one spot to fill in, and then send the message.

Scoring:

Now we'll see what our errors look like.

You can see here, there is a total score of 125, that meant they were at 50 each, two aircraft holds, and you saw both of those when they hit the yellow line, both outgoing planes in this case.

At one penalty point each per minute for a welcome delay, it took me 5 minutes to get that aircraft welcomed, and when the plane turned red, it took me 2 minutes to get him out of red, because that's 10 penalty points per minute for each holding delay.

You can also see here there were no duplicate messages, no extraneous clicks – those would be when you click on the air traffic controller when it's not required for example -- and no incorrect messages – those going out were transferred and those coming in were accepted.

["Dual Task Demo 1 no delays"]
FULL ***DUAL Demo 1 – 3 planes:***
do color perfect;
accept all altitude requests;
get one altitude request error,
no altitude delays

In this demonstration, I'm going to add the altitude change request task to the hand-off task you've already seen. I will do all the hand-off tasks correctly.

You'll see here that this plane has just turned magenta, it's requesting an altitude change; since we don't know what will be correct, so for this demo, I'll just accept all the requests.

This was correct, see the smiley face, and hear the high tone. In the mean time, this incoming plane has turned green and is requesting to be accepted in, so I click Accept, I click plane, the Air Traffic Controller, and I send the message.

This plane that had the altitude request, has now turned brown and is asking to go out, so I will click the Transfer Action Button, the airplane and Air Traffic controller, as usual, and send the message.

This plane now is asking for an altitude request, so I will accept that too. You heard there was a low tone and you see the big X that tells us that our selection was incorrect, we should have rejected this one.

This one now is saying hello in the usual fashion, so I will welcome this plane in and send him on his way.

This one has been accepted by East and has turned yellow, so I'm going to tell him to Contact East, and send this message along as well.

So now I'm waiting to see if there are any other altitude requests, and what other business I have to attend to. Yes, he has just reached the green line and turned brown, so I'm going to transfer this plane to the Northern sector and send this message along.

Right now there is no other business that needs attending to.

North has told us that it accepting this plane, but I'd better take care of this altitude request first, because there is more of a time issue on that, and I'm going to accept this request and send this message, and see it is correct – see the Smiley Face and hear the high tone, and this one I can now tell to Contact North and Send the message.

Scoring:

As you can see here on the scoring, there is a total score of 100, and that was because of the altitude request that I made in error that cost me 100 points. But, to ignore a request will cost 200 points for sure, so it's always better to guess. You may be right, but if you're wrong, it will be only 100 points, not 200.

["Dual Task Demo 2 with delays"]
FULL ***DUAL Demo 2 – 3 planes:***
do color perfect;
reject all altitude requests;
get one altitude request error,
Do not respond to first altitude request
Let next two begin blinking and then respond in time

This time I'm going to demonstrate the altitude change request task plus the hand-off task again, but I'll delay in responding to the altitude requests to show you what happens when I do that. When I do handle an altitude request, I'll reject them all for this demo.

Again you'll see we have the three planes, and this one had turned magenta is asking for an altitude request. I think I'll just ignore him altogether, so I'm going to let him continue on his way and you'll see what happens.

He is now blinking, and you will notice all the action buttons up here have been grayed out. I can't handle anything, including this incoming plane, until I take care of him. But, if I let him continue on until he stops blinking, he'll go back to white, and these buttons will be returned to my use, but I will have missed my opportunity to respond to the Altitude change request. I will now Accept this airplane coming from the West and Send the message.

And, he's now exiting, so I'll start the Transfer procedure for him. As you see here in the text box, there was no altitude request handled.

Now, he is moving along, but again, I am going to delay this, and let him start blinking so that you'll again these buttons gray-out; there he goes, [beep sounds], you heard that beep as well to draw your attention to this, so for this demo, I'm going to reject all the requests this time. So I'll reject that and it gives me back my Action buttons, and did you see that Smiley Face right there and hear the high tone, that tells you it was a correct response.

I'm now going to have this plane contact the Eastern sector, and I have a chance to Welcome this one in as well. He's now going North, and he's requesting a Transfer. Again, we're just waiting here for these planes to reach a point where we have to do something.

This plane now is requesting an Altitude change; I'm going to delay responding to that too. In the mean time I can still handle him, because the Altitude change request hasn't started blinking yet, so I'll have him do his Contact and Send the message. There he goes blinking, and I can't do anything but handle his request. I will Reject his request as well and click the Reject button, the airplane, you'll see in the template there is only one spot to fill, and Send the message. Oh, no, it was wrong – you see the X and hear the low tone? So, that was an incorrect choice.

Scoring:
And, as you see on the scoring here, there are a total of 300 points. That one plane that I ignored, cost me 200 points, and the one error I made was only 100 points, that was the altitude change request that I rejected and should have accepted, the plane that gave me the X and the low tone. So, you see it is worth your while to make a guess even if you are not sure whether to accept or reject an altitude change request, because not responding is much more expensive, not responding costs 200 points, and getting an error message is only 100.

Coached Dual:

Now we're going to start a trial that will have both the color task and the altitude request task, and I will coach you through this one. This will be one that you will do. I will make sure you get all the color part correct, and I will help you with the procedure when it comes to the Altitude request, but I will let you make the decision for the Altitude request, as to whether to Accept or to Reject the request.

["Primary Task Demo 1 no delays"]
CONTROL DEMO 1 with 3 white and 3 magenta – no delays

In this demonstration, I will show how to handle the Altitude Change Request task. As you see there are three planes moving on the screen. Two are leaving my sector, and one is entering. My sector is that area bounded by the yellow border. I am the air traffic controller in the center.

This plane has turned magenta. So, I'm going to Accept this altitude change request by pressing the Accept Altitude Request Action Button, the airplane, and Sending the message. And you'll notice there is the Smiley Face and the high tone saying I actually did the correct thing. You will all see in this text box what the plane requested and what I actually responded with. These are recorded here in case you want to go look back and see what was done. For this demo, I will be accepting all the requests.

Now this first plane is moving along, and we're waiting for another to turn magenta, to again demonstrate what to do. *[next plane:]* When you are responding to Altitude requests, you always start with the action button, click on the airplane, and Send the message along. You see that this one had the X with the low tone; that means that in this case we should have rejected the altitude request.

Again, the planes are moving and continuing through the area of your control.

Now, you see this plane has turned magenta, and we're going to be accepting all the requests for this demo, so we'll say Accept, click on this airplane, and Send the message away, and you see that this was a correct response again, because the Smiley Face and high tone came on.

That's it for this and you'll see what the scores come up and say.

Scoring:

Now you notice on this score card that there was one altitude request error of 100 points – that was the one you saw that had the X and the low tone. And, we did respond to everybody, so there is no penalty for No Responses.

Handle 1st plane – mention feedback - sight and sound
Mention – "accept all"
Handle 2nd plane – mention feedback - sight and sound
Mention properties coming and going
Mention text area

["Primary Task Demo 2 with delays"]
CONTROL DEMO 2 with 3 white and 3 magenta – 1 NR + 2 delays

In this demonstration, I will show you what happens when I delay in responding to an Altitude Change Request or do not respond at all.
You will see that this airplane has already turned magenta, and I'm going to ignore this one and show you what happens. As this plane starts blinking, you hear the little beep to draw your attention to it. The beep and the blinking are telling you that there is little time remaining to Accept or Reject this plane's request before you lose the opportunity to do so. It has now become urgent for you to act. Now it has stopped blinking and has returned to white and we have not responded and you'll see what that costs when we see the scores.

Now we're watching for the next plane to turn magenta. This one now has turned magenta, and during this demo we'll Reject all the Altitude Change requests, but we'll wait a moment until that one starts blinking, and then we'll Reject it. There's the beep to get your attention, and it's blinking. So we'll click the Reject Altitude Request button, the aircraft and Send the message. And, this one was a correct response for that request, you can tell by the Smiley Face here and the high tone.

Now we're waiting for the third plane to request an altitude change. See now, this plane has requested an Altitude change, the request shows up in that little box down here, and we're again going to wait so you can experience the beep and blinking here. There is the beep and blink, so we're going to reject this request, we're going to click on the aircraft and Send the message along. And that was an incorrect one, we should have Accepted that one – you know that because it gave you and X and a low tone.

Scoring:

Here are the scores for this one. As you can see, there is a total of 300 points. 100 of those were for the incorrect Altitude Request response, and you'll see there are 200 points for not responding to that very first plane. So, it's very, very much in your interest to respond even if you are not sure whether to Accept or Reject the Altitude Change request, because if you don't respond at all, you get 200 penalty points. If you at least respond, you get only 100 points if you make an incorrect choice – and your choice may even be correct.

Coached Control Practice:

Now, I'm going to click the score screen away and give you an opportunity to try this yourself. I'll be coaching you along. I won't give you any information whether to accept or reject the Altitude Change request, you'll have to make that decision yourself, but I will make sure you are clicking the buttons in the correct order.

Scoring (of coached practice):

Now you see the score here. The total score is 490 points. Your altitude request error score is 200 – that means there were two incorrect Altitude request choices – or two planes with an X and a low tone. Since you did indeed respond to all the requests, there were no penalties for No Response.

A.4 Debrief Protocol and Analysis

A.4.1 Debrief Protocol

Note: The items followed by a (3) refer to the transfer test. All other items refer to the main experiment.

Categ.	Cond.	Sub.	Question #	Item	Value
3	Dual-Low	SUB1	1	How did you decide whether to accept or reject an altitude change request?	because of the amount of turbulence?
3	Dual-Low	SUB1	1	Did your strategy change over the 8 trials? [Please explain]	yes, I thought it might be the direction, the area it was in, the location of nearby planes and the amount of fuel for the size of the plane
3	Dual-Low	SUB1	1	How did you decide whether to accept or reject an altitude change request? (3)	I completely guessed.
3	Dual-Low	SUB1	2	Did you use a rule?	Yes
3	Dual-Low	SUB1	2	If Yes, I accepted an altitude request when	the fuel, plane size, and turbulence was opposite
3	Dual-High	SUB2	1	How did you decide whether to accept or reject an altitude change request?	I used the past answers that I had done before and went from there based on what I had remembered as correct and incorrect.
3	Dual-High	SUB2	1	Did your strategy change over the 8 trials? [Please explain]	at first I was just guessing with no help then I started to notice a pattern and tried to remember the size and the turbulence combinations that were correct or incorrect.
3	Dual-High	SUB2	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to compare the numbers with the previous ones I had learned. I made partially educated guesses.
3	Dual-High	SUB2	2	Did you use a rule?	Yes
3	Dual-High	SUB2	2	If Yes, I accepted an altitude request when	The size was small and the turbulence was 1. Also when the size was large with a low turbulence and a low gas percent. When the turbulence was 3 for the

					small planes I also accepted.
3	Dual-High	SUB2	2	If No, what did you do?	the large plane with a high gas percentage and a higher turbulence was rejected
3	Control	SUB3	1	How did you decide whether to accept or reject an altitude change request?	I based my answer first upon the percentage of gas remaining in the aircraft. If it was 40%, then I used the airplane size to determine whether to accept or deny. Large I rejected and small I accepted. If the percentage was 20, I looked at the level of turbulence. If it was high, I rejected, low I accepted.
3	Control	SUB3	1	Did your strategy change over the 8 trials? [Please explain]	The first and half of the second trial was mainly trial and error. After that, I used my strategy above.
3	Control	SUB3	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to use the knowledge I had gained in the first 8 trials by basing my answer first by looking at the percentage of gas remaining. The difficulty was mainly when deciding about crafts with 30% remaining. I used an average between the statistics I used to determine accepting or rejecting 20 and 40% aircrafts. I did not accept a plane with 30% fuel that was larger than medium sized or had turbulence above 2.
3	Control	SUB3	2	Did you use a rule?	Yes
3	Control	SUB3	2	If Yes, I accepted an altitude request when	20% planes had a turbulence level of 1 and when 40% planes were size S.
6	Dual-High	SUB4	1	How did you decide whether to accept or reject an altitude change request?	I based my answers upon whether the plane had enough fuel to make an altitude change and upon how rough the turbulence was for the plane.
6	Dual-High	SUB4	1	Did your strategy change over the 8 trials? [Please explain]	My strategy was more consistent in the last trial because it seemed as if it was more effective during the last trial.

6	Dual-High	SUB4	1	How did you decide whether to accept or reject an altitude change request? (3)	I based my judgment upon the same factors although it was a bit more difficult to decide as more information was given. I made judgments as logically as possible although I had no idea what exactly I was trying to avoid or avert.
6	Dual-High	SUB4	2	Did you use a rule?	Yes
6	Dual-High	SUB4	2	If Yes, I accepted an altitude request when	I tried to accept requests when fuel was low and turbulence was high. This was not always effective.
6	Dual-Low	SUB5	1	How did you decide whether to accept or reject an altitude change request?	I looked at the size of the plane and the turbulence. If it was a larger plane with a larger amount of turbulence, I accepted the request.
6	Dual-Low	SUB5	1	Did your strategy change over the 8 trials? [Please explain]	This strategy seemed to only work in a certain number of trials. In some trials, I didn't have any strategy at all because nothing seemed to work.
6	Dual-Low	SUB5	1	How did you decide whether to accept or reject an altitude change request? (3)	Again I looked at the size of the plane and the turbulence facto but this time I also added in the factor of the fuel. If they were all relatively high I accepted the request. If they weren't, I didn't.
6	Dual-Low	SUB5	2	Did you use a rule?	No
6	Dual-Low	SUB5	2	If No, what did you do?	I picked them randomly after seeing that my strategy didn't really work.
6	Control	SUB6	1	How did you decide whether to accept or reject an altitude change request?	During the first part of the task, I decided how to accept or reject them by memorization.
6	Control	SUB6	1	Did your strategy change over the 8 trials? [Please explain]	No, it did not.
6	Control	SUB6	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to remember what I decided for the first 8 trials and estimate the new dimensions with what I used before. If I was given 10 XS 1 and during the ninth trial was given 20 XS 1, I would associate those as very similar and give the same answer as I had for 10XS1. I did this throughout

					the entire 9th trial trying to remember what I had answered in the previous 8 trials.
6	Control	SUB6	2	Did you use a rule?	Yes
6	Control	SUB6	2	If Yes, I accepted an altitude request when	I based the majority of what I had done on memorization, however I also noticed a pattern of opposites. For example if 40 L 1 was always incorrect, 20 S 1, was always correct. The 20 and 40 being on opposite scales. However, if I had more time, I would have most likely paid more attention to what these opposites were and if they were all laid out in front of me.
1	Dual-Low	SUB7	1	How did you decide whether to accept or reject an altitude change request?	I noticed that regardless of the percent or size of the plane, when the turbulence was 1, the request should have been accepted, and when the turbulence was 3, the request should have been rejected.
1	Dual-Low	SUB7	1	Did your strategy change over the 8 trials? [Please explain]	No. After the first trial I noticed that this strategy worked, and so I continued to follow it for the remaining 7 trials.
1	Dual-Low	SUB7	1	How did you decide whether to accept or reject an altitude change request? (3)	I figured that since turbulence seemed to be the determining factor, and that a high turbulence should be rejected while a low one should be accepted, I applied this to the last trial. However, a turbulence of 2 falls in the middle of the range, in which case I mostly guessed as to whether to reject or accept the request.
1	Dual-Low	SUB7	2	Did you use a rule?	Yes
1	Dual-Low	SUB7	2	If Yes, I accepted an altitude request when	the turbulence was 1
1	Dual-High	SUB8	1	How did you decide whether to accept or reject an altitude change request?	Throughout the first few trials I just guessed, during the last trials....If the fuel was on the higher side for

					medium to larger (medium to low for smaller planes) planes and the turbulence was low I accepted the request. If the fuel was medium to low, the plane medium to larger and the turbulence on the medium to higher side I rejected it.
1	Dual-High	SUB8	1	Did your strategy change over the 8 trials? [Please explain]	At first I was simply guessing, by the seventh or eighth trial I recognized a pattern.
1	Dual-High	SUB8	1	How did you decide whether to accept or reject an altitude change request? (3)	I applied what I used in the last few of the first eight trials. (See question 1
1	Dual-High	SUB8	2	Did you use a rule?	Yes
1	Dual-High	SUB8	2	If Yes, I accepted an altitude request when	The plane had enough fuel for the size of the plane and the turbulence was relatively low.
1	Control	SUB9	1	How did you decide whether to accept or reject an altitude change request?	I rejected if the turbulence level was 3 and accepted if it was 1.
1	Control	SUB9	1	Did your strategy change over the 8 trials? [Please explain]	Yes, my guesses were random at first and during the second trial I noticed the pattern.
1	Control	SUB9	1	How did you decide whether to accept or reject an altitude change request? (3)	I again based my decisions on the turbulence level. If it was 1 or less I accepted the request. If it was 2 or more I rejected it. I wasn't sure about the 2 but I decided it would be safer to reject than to accept.
1	Control	SUB9	2	Did you use a rule?	Yes
1	Control	SUB9	2	If Yes, I accepted an altitude request when	the turbulence level was 1.
3	Dual-High	SUB10	1	How did you decide whether to accept or reject an altitude change request?	I noticed a pattern between the numbers and letters at the bottom of the plane... I would recognize the pattern to either reject or accept the altitude request.
3	Dual-High	SUB10	1	Did your strategy change over the 8 trials? [Please explain]	No. My strategy became more noticeable and better with more experience
3	Dual-High	SUB10	1	How did you decide whether to accept or reject an altitude change request? (3)	I had absolutely no idea and randomly selected answers...I applied no strategy whatsoever.
3	Dual-High	SUB10	2	Did you use a rule?	Yes
3	Dual-High	SUB10	2	If Yes, I accepted an altitude request when	20 and 3 were rejected as well as 40 and 3... When

					the numbers would appear I followed my instinct...and also tried to keep the above pattern with minor alterations.
3	Dual-Low	SUB11	1	How did you decide whether to accept or reject an altitude change request?	40 L 1 20 L 1 20 S 1 20 S 3 These were the only ones to be accepted.
3	Dual-Low	SUB11	1	Did your strategy change over the 8 trials? [Please explain]	No
3	Dual-Low	SUB11	1	How did you decide whether to accept or reject an altitude change request? (3)	I rejected almost all the new ones, except ones that had an even number followed by an even number at the end.
3	Dual-Low	SUB11	2	Did you use a rule?	Yes
3	Dual-Low	SUB11	2	If Yes, I accepted an altitude request when	The information matched the following: 40 L 1 20 L 1 20 S 1 20 S 3
3	Control	SUB12	1	How did you decide whether to accept or reject an altitude change request?	Tested different patterns until I found patterns that yielded in a smiley face.
3	Control	SUB12	1	Did your strategy change over the 8 trials? [Please explain]	Only changed from testing patterns to implementing the pattern that worked.
3	Control	SUB12	1	How did you decide whether to accept or reject an altitude change request? (3)	If the plane was L or above & the turbulence was 3 or above I rejected. If the plane was S or lower & the fuel was 20% or lower I rejected. If the plane was M I rejected if either turbulence or fuel % were towards the ends of the scale, basically if either was far from the middle I rejected.
3	Control	SUB12	2	Did you use a rule?	Yes
3	Control	SUB12	2	If Yes, I accepted an altitude request when	If plane was L & turbulence was 3 I rejected. If plane was L & turbulence was 1 I accepted. If plane was S & fuel was 40% I rejected. If plane was S & fuel was 20% I accepted.
6	Dual-Low	SUB13	1	How did you decide whether to accept or reject an altitude change request?	I decided whether to accept or reject an altitude change based upon the decisions that enabled me to receive smiley faces in the previous trials.
6	Dual-Low	SUB13	1	Did your strategy change over the 8 trials? [Please explain]	Not really because during all eight trials, I was still trying to figure out which decisions would provide me

					with the smiley faces.
6	Dual-Low	SUB13	1	How did you decide whether to accept or reject an altitude change request? (3)	I rejected all of the choices that contained any of the new options since in the directions that were given to me, it was stated that the things that I learned in the previous trials still counted. I tried to go on what appeared familiar to me from the previous trials although I still was not completely sure.
6	Dual-Low	SUB13	2	Did you use a rule?	Yes
6	Dual-Low	SUB13	2	If Yes, I accepted an altitude request when	When a choice appeared that I believed that in previous times when accepted provided me with the smiley faces.
6	Dual-High	SUB14	1	How did you decide whether to accept or reject an altitude change request?	I remembered which combinations had received a smiley face in the past and which had received an X
6	Dual-High	SUB14	1	Did your strategy change over the 8 trials? [Please explain]	no, as the trials went on I remembered better which combinations were for "accept" and which were for "reject" (there weren't that many possible combinations
6	Dual-High	SUB14	1	How did you decide whether to accept or reject an altitude change request? (3)	I likened the XS planes to the S planes and the XL planes to the large and used the same strategy, remembering which were accepted before and using the lower end of the %fuel and turbulence for 20 and 1 respectively, and the higher for 40 and 3. I guessed for the medium planes.
6	Dual-High	SUB14	2	Did you use a rule?	No
6	Dual-High	SUB14	2	If No, what did you do?	I just remembered I didn't really form any concrete rules in my head, just recognition.
6	Control	SUB15	1	How did you decide whether to accept or reject an altitude change request?	The ones that I accepted were 40 L 3, 40 S 1, 20 L 1, and 20 S 3. I figured that the planes with 40% fuel left should have a change in altitude if both the size of plane was large and there was high turbulence, or if

					the size of the plane was small and there was low turbulence. The opposite was true for the planes with 20% fuel left, that is, they should be accepted for change in altitude if large size with low turbulence, or small planes with high turbulence.
6	Control	SUB15	1	Did your strategy change over the 8 trials? [Please explain]	No, once I figured out that pattern worked I stuck with it through the 8 trials.
6	Control	SUB15	1	How did you decide whether to accept or reject an altitude change request? (3)	I used the same reasoning that I used for the initial trials. For the planes with larger amounts of fuel left, such as 50%, I figured that if they were L or XL in size should be accepted for altitude change with higher turbulence, such as 3 or 4, and vice versa if they were XS or S (with turbulence of 0 or 1). I also figured that if the plane was medium size to accept it with a turbulence of 2. Once again I used the opposite approach for planes with smaller amounts of fuel, such as 10% (smaller size and higher turbulence or larger size and lower turbulence). For planes with 30% of fuel I accepted all requests for turbulence.
6	Control	SUB15	2	Did you use a rule?	Yes
6	Control	SUB15	2	If Yes, I accepted an altitude request when	The plane had 40% left and had either large size and high turbulence or small size and low turbulence, and for 20% had either large size and low turbulence or small size and high turbulence (thus 40% had the same high numbers/sizes while 20% had opposites).
1	Dual-High	SUB16	1	How did you decide whether to accept or reject an altitude change request?	On the first trial, I saw that planes with 40% of fuel received an X when I accepted the altitude change.
1	Dual-High	SUB16	1	Did your strategy change over	No. I followed that pattern

				the 8 trials? [Please explain]	and most of my answers were correct.
1	Dual-High	SUB16	1	How did you decide whether to accept or reject an altitude change request? (3)	Based on the previous trials, I followed the same pattern of not accepting altitude changes for flights with 40% or more fuel.
1	Dual-High	SUB16	2	Did you use a rule?	Yes
1	Dual-High	SUB16	2	If Yes, I accepted an altitude request when	When the fuel percentage was lower than 40%.
1	Dual-Low	SUB17	1	How did you decide whether to accept or reject an altitude change request?	For the last 8 trials, 20 were accepted and 40 were rejected, I figured this out by trial and error during the first trial. I guessed twice, made an assumption and tested it out. It turned out to be right so I went with it.
1	Dual-Low	SUB17	1	Did your strategy change over the 8 trials? [Please explain]	Well, after I figured out that my guess was right during the first trial, I just stuck with that.
1	Dual-Low	SUB17	1	How did you decide whether to accept or reject an altitude change request? (3)	Since 20 were accepted I assumed that 10 would be too. Since 40 were rejected I assumed that 50 would be too. 30 was the only one I questioned. Before I began, I was thinking about determining 30 based on the size of the plane, but for all the 30% ones, the size was medium so that didn't really help me. Finally, I just assumed that the greater the turbulence, the greater the need for the plane to change altitude so I went with that. (I'm not sure if that was a correct assumption or not).
1	Dual-Low	SUB17	2	Did you use a rule?	Yes
1	Dual-Low	SUB17	2	If Yes, I accepted an altitude request when	Yes, I accepted all the planes at 20% and denied all the planes at 40% (for trial 9, my rule was that I accepted 10 and 20 and I only accepted 30 when the turbulence was 4 or 5
1	Control	SUB18	1	How did you decide whether to accept or reject an altitude change request?	Accept if the plane's fuel was at 20, reject at 40
1	Control	SUB18	1	Did your strategy change over the 8 trials? [Please explain]	Not once I figured out that 20 were repeatedly accepted-- I think that may

					have been in the second trial.
1	Control	SUB18	1	How did you decide whether to accept or reject an altitude change request? (3)	I only accepted those with a fuel tank value of 20 as in the past and disregarded all the new extraneous information
1	Control	SUB18	2	Did you use a rule?	Yes
1	Control	SUB18	2	If Yes, I accepted an altitude request when	the fuel percentage was 20
3	Dual-Low	SUB19	1	How did you decide whether to accept or reject an altitude change request?	mainly based on the turbulence, then the fuel amount left. I also took into consideration the direction the plane was moving.
3	Dual-Low	SUB19	1	Did your strategy change over the 8 trials? [Please explain]	Yes. I realized that regardless of the plane size and fuel left, sometimes the direction of the plane determined whether to accept or reject altitude changes. So I began to memorize which way the answers didn't make sense.
3	Dual-Low	SUB19	1	How did you decide whether to accept or reject an altitude change request? (3)	Basically, the ones with different numbers than previous I guessed based on the knowledge I had gathered before such as direction and fuel to size ratio
3	Dual-Low	SUB19	2	Did you use a rule?	Yes
3	Dual-Low	SUB19	2	If Yes, I accepted an altitude request when	most of the time I tried to stick to the idea that when the plane has a sufficient amount of fuel for the size of the plane, the turbulence played the least role of determining whether to accept or reject the altitude change
3	Dual-High	SUB20	1	How did you decide whether to accept or reject an altitude change request?	I rejected all of the planes that did not have an "s" except for 40 L 3. I accepted all planes that did have an "s" except for 20 S 3.
3	Dual-High	SUB20	1	Did your strategy change over the 8 trials? [Please explain]	No.
3	Dual-High	SUB20	1	How did you decide whether to accept or reject an altitude change request? (3)	I applied the same rules to the last task as I did to the earlier tasks.
3	Dual-High	SUB20	2	Did you use a rule?	Yes
3	Dual-High	SUB20	2	If Yes, I accepted an altitude request when	the plane had an "s" and was not 20 S 3 and when

					the plane was 40 L 3.
3	Control	SUB21	1	How did you decide whether to accept or reject an altitude change request?	I realized that any plane with 20 and 3, regardless of S or L should be rejected. Also, any plane with 40 and 3 should be accepted. In terms of planes with turbulence 1, if it was S it should be accepted and if it were L, rejected, regardless of fuel.
3	Control	SUB21	1	Did your strategy change over the 8 trials? [Please explain]	It took me a few trials to realize how to decide whether to accept or reject, but once I came up with the above strategy, I kept to it.
3	Control	SUB21	1	How did you decide whether to accept or reject an altitude change request? (3)	I wasn't completely certain at any point during the last task. I tried to reason according to the first strategy I came up with. Planes below 20, and also below 3, I believed should be rejected. Accordingly, those about 40, and also above 3, I believed should be accepted. However, when they fell in between, I was uncertain. I attempted to reason with one aspect first...either the fuel or turbulence, according to the original strategy...and then in a way, guessed at the final aspect. I did not know if the original strategy was effective here.
3	Control	SUB21	2	Did you use a rule?	Yes
3	Control	SUB21	2	If Yes, I accepted an altitude request when	the fuel was at 40 and the turbulence was 3, regardless of size, and if the turbulence was 1, and the size of the plane was S, regardless of amount of fuel.
6	Dual-High	SUB22	1	How did you decide whether to accept or reject an altitude change request?	I decided based on the previous trails.
6	Dual-High	SUB22	1	Did your strategy change over the 8 trials? [Please explain]	No. Not really because it's was stated that my same expectations should be the same throughout the trails.
6	Dual-High	SUB22	1	How did you decide whether to accept or reject an altitude change request? (3)	I thought about each one's individual properties and then I decided from there.

6	Dual-High	SUB22	2	Did you use a rule?	No
6	Dual-High	SUB22	2	If No, what did you do?	I tried to memorize the responses however; I couldn't keep up because I had other tasks to keep track of.
6	Dual-Low	SUB23	1	How did you decide whether to accept or reject an altitude change request?	at first, it was just random -- then I started to see a pattern and messed around with it until it seemed to work
6	Dual-Low	SUB23	1	Did your strategy change over the 8 trials? [Please explain]	yes, once I started to see a pattern I changed it here and there until it worked, after I got the pattern I simply used it for the remaining trials
6	Dual-Low	SUB23	1	How did you decide whether to accept or reject an altitude change request? (3)	if the planes had 20% or 40% fuel -- I knew how to handle those for the most part but the other ones I had no clue so I just guessed
6	Dual-Low	SUB23	2	Did you use a rule?	Yes
6	Dual-Low	SUB23	2	If Yes, I accepted an altitude request when	40 L 3, 40 S 1, 20 L 1, and 20 S 3
6	Control	SUB24	1	How did you decide whether to accept or reject an altitude change request?	generally, I chose accept if the turbulence was low. However, if the fuel was low and the plane was large, I would choose to reject
6	Control	SUB24	1	Did your strategy change over the 8 trials? [Please explain]	it took me several rounds to discover the importance of the turbulence rating. Before I discovered this, I paid more attention to fuel and size.
6	Control	SUB24	1	How did you decide whether to accept or reject an altitude change request? (3)	I used the same strategy, treating XS as small and XL as large. With the medium planes, I guessed
6	Control	SUB24	2	Did you use a rule?	Yes
6	Control	SUB24	2	If Yes, I accepted an altitude request when	The plane had a low turbulence rating. If the plane was large, however, it was only granted if the fuel was 20 or 10.
1	Dual-Low	SUB25	1	How did you decide whether to accept or reject an altitude change request?	I saw a pattern that the L planes were rejected and the small were accepted, so I followed in this pattern.
1	Dual-Low	SUB25	1	Did your strategy change over the 8 trials? [Please explain]	Yes. Initially I was not able to identify any sort of pattern. Through trial and

					error I arrived at the pattern and based my choices on that. So, about three trials in I understood which planes would be accepted and which would be rejected.
1	Dual-Low	SUB25	1	How did you decide whether to accept or reject an altitude change request? (3)	Based on what I previously saw in the prior trials, the large planes were rejected and the small were accepted. So, in the last trial I followed in this pattern. When presented with a XS or XL plane I also continued in this same pattern. When I was presented an M plane I used my judgment based on the other factors (fuel and turbulence). Less fuel and higher turbulence I would reject and more fuel, lower turbulence I would accept.
1	Dual-Low	SUB25	2	Did you use a rule?	Yes
1	Dual-Low	SUB25	2	If Yes, I accepted an altitude request when	When The plane was marked L.
1	Dual-High	SUB26	1	How did you decide whether to accept or reject an altitude change request?	The request was based on the size of the plane, with smaller planes getting approval and larger planes being denied.
1	Dual-High	SUB26	1	Did your strategy change over the 8 trials? [Please explain]	Over the first 8 trials, I made my decisions only on account of the plane size.
1	Dual-High	SUB26	1	How did you decide whether to accept or reject an altitude change request? (3)	Requests were approved for smaller planes with low fuel level and high turbulence. Requests were denied for larger planes with more fuel and low turbulence. The logic was that a larger, heavier plane (with more fuel) would be able to better handle higher turbulence than a smaller, lighter plane.
1	Dual-High	SUB26	2	Did you use a rule?	Yes
1	Dual-High	SUB26	2	If Yes, I accepted an altitude request when	The airplane was S (small).
1	Control	SUB27	1	How did you decide whether to accept or reject an altitude change request?	from the size of the plane
1	Control	SUB27	1	Did your strategy change over	yes at first I would just try

				the 8 trials? [Please explain]	yes or no at random but then I always chose to accept the smaller planes altitudes requests as the trials went on
1	Control	SUB27	1	How did you decide whether to accept or reject an altitude change request? (3)	if the plane was XS S or M sized I accepted the request, if it was L or XL I denied it
1	Control	SUB27	2	Did you use a rule?	Yes
1	Control	SUB27	2	If Yes, I accepted an altitude request when	the size of the plane was small
3	Dual-High	SUB28	1	How did you decide whether to accept or reject an altitude change request?	I tried to look at patterns and remember what had been accepted and not accepted prior.
3	Dual-High	SUB28	1	Did your strategy change over the 8 trials? [Please explain]	In the beginning I looked for more of a pattern i.e. are all smalls accepted or are all 20's rejected, etc. But then I realized that it seemed that I just needed to remember what was approved and what was rejected. i.e.: all 20 L 3 were OK.
3	Dual-High	SUB28	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to apply my prior strategy by rounding up and down, using math in a way. If there was a 10 for example, I generalized it to a 40 instead of a 20 and went from there. If I had an O I thought of it as a 1 in the previous episodes, etc...
3	Dual-High	SUB28	2	Did you use a rule?	No
3	Dual-High	SUB28	2	If No, what did you do?	memorized which combinations were associated with the happy faces and which were associated with a growl.
3	Dual-Low	SUB29	1	How did you decide whether to accept or reject an altitude change request?	If it had a 40 and a 1, it was rejected. If it had a 20 and an L, it was accepted. Besides that, you accepted 40 L 3 and 20 S 1.
3	Dual-Low	SUB29	1	Did your strategy change over the 8 trials? [Please explain]	Not really, just continuous process of elimination.
3	Dual-Low	SUB29	1	How did you decide whether to accept or reject an altitude change request? (3)	The two exceptions that were accepted in the 8 trials, 40 L 3 and 20 S 1, the last two matched in corresponding order. So sometimes I went by that,

					and other times I went by if anything else matched.
3	Dual-Low	SUB29	2	Did you use a rule?	Yes
3	Dual-Low	SUB29	2	If Yes, I accepted an altitude request when	There was a 20 with an L, there was a 40 L 3, or there was a 20 S 1.
3	Control	SUB30	1	How did you decide whether to accept or reject an altitude change request?	The planes that had all the lowest descriptions together or the highest descriptions all together were accepted. The large planes coupled with twenty percent were all accepted, no matter what the turbulence. The small planes with a fuel percentage of twenty were rejected at high turbulence, and large planes at forty percentage of fuel were rejected at low turbulences.
3	Control	SUB30	1	Did your strategy change over the 8 trials? [Please explain]	The first trial was pure guessing with a little influence from the practice ones even though it was said that they could be different. Then with each trial, I gradually weeded out some responses until becoming sure of each one. I used tricks to remember them when they first became more familiar in the early trials, and then they eventually became memorized for about the last three or four trials.
3	Control	SUB30	1	How did you decide whether to accept or reject an altitude change request? (3)	The descriptions of the planes that were exactly the same as in the previous eight trials received the same answers. Like in question one; I accepted all planes that had all the smallest descriptions and all the highest descriptions grouped. The planes that fell in the middle categories were more guesswork, but I tried to reject or accept based on how far toward either extreme the descriptions were. I was fairly unsure of decisions that were not identical to

					the first eight trials due to the lack of responses.
3	Control	SUB30	2	Did you use a rule?	Yes
3	Control	SUB30	2	If Yes, I accepted an altitude request when	The planes were either grouped by their extremes (all high together or all low together) and when a large plane was coupled with a twenty percentage of fuel it was accepted regardless of turbulence.
6	Dual-Low	SUB31	1	How did you decide whether to accept or reject an altitude change request?	I didn't have a method...guessing
6	Dual-Low	SUB31	1	Did your strategy change over the 8 trials? [Please explain]	I kept changing the strategy but never found a pattern
6	Dual-Low	SUB31	1	How did you decide whether to accept or reject an altitude change request? (3)	Since I didn't find a pattern from before to apply, I used my own judgment (i.e. if the plane was experiencing moderate to severe turbulence and had 30% fuel or more, I accepted
6	Dual-Low	SUB31	2	Did you use a rule?	No
6	Dual-Low	SUB31	2	If No, what did you do?	Guessed
6	Dual-High	SUB32	1	How did you decide whether to accept or reject an altitude change request?	Based on the combination of the three factors - if it was a small plane with lots of fuel, I accepted it; if it was a large plane with lots of turbulence and not a lot of fuel, I rejected it.
6	Dual-High	SUB32	1	Did your strategy change over the 8 trials? [Please explain]	Yes - the first trial was random, and then I switched to accepting all requests because I couldn't figure out the pattern, and then I switched to the strategy above.
6	Dual-High	SUB32	1	How did you decide whether to accept or reject an altitude change request? (3)	Same as above, based on the combination of the three factors.
6	Dual-High	SUB32	2	Did you use a rule?	No
6	Dual-High	SUB32	2	If No, what did you do?	I used the combination of all three factors, but my decision was influenced by whether or not the pattern I was deciding on had been marked correct or not the last time I had decided on that pattern - for instance, I noticed that 20-S-1 was always supposed to be rejected, even though I would have accepted it

					based on the fact that it had some fuel left, was small, and there wasn't much turbulence.
6	Control	SUB33	1	How did you decide whether to accept or reject an altitude change request?	By finding a pattern between the altitude and the fuel in relation to the size of the plane, a pattern was formed that I was able to recognize after about the 4th trial. Before that, my basis of accepting and rejecting was based on trial and error, as well as grouping of similar information in relation to the response I received from my answer.
6	Control	SUB33	1	Did your strategy change over the 8 trials? [Please explain]	Yes, at first I used trial and error. Second, I used groupings of fuel or plane size and altitude, to trial and error those results. Once a grouping was proved wrong, I attempted to make another grouping theory and follow that theory until proved wrong or right.
6	Control	SUB33	1	How did you decide whether to accept or reject an altitude change request? (3)	I based my answers on what I realized was correct in the previous trials, while taking into account the added information. I saw a cross pattern with regards to the small planes and a straight, streamline pattern with the larger planes. When deciding to accept or reject, I took into account the way each property would look lined up on paper. From that point I would determine if there was a cross pattern or straight pattern occurring. If neither existed, I would reject the request. If one of the patterns existed I would accept the request Only IF there was a symmetrical cross pattern or straight pattern. All other slight changes from this pattern were rejected. For example: small planes were

					cross patterned by a high fuel and low turbulence or a low fuel and a high turbulence (20 S 3 or 40 S 1); large planes on the other hand, had straight/ high to low patterns (40 L 3 or 20 L 1) with respect to fuel and turbulence. With regards to the final trial where new information was added (fuel 10-50 for example), if a straight pattern existed (10 XS 0 or 30 M 2) I accepted these based on the large plane pattern I had encountered in the first 8 trials. With regards to the cross pattern seen in the small planes, I would accept if there was a symmetrical cross between information (10 M 4 or 20 s 3 or 50 L 1 for example).
6	Control	SUB33	2	Did you use a rule?	Yes
6	Control	SUB33	2	If Yes, I accepted an altitude request when	Large planes were at high fuel and high turbulence or low fuel and low turbulence. Small planes were accepted at low fuel and high turbulence or high fuel and low turbulence. All other requests that did not fit into either pattern were rejected.
1	Dual-High	SUB34	1	How did you decide whether to accept or reject an altitude change request?	the first two that I tried I based on size. The small I accepted and the large I rejected. they were both right so I based everything on that
1	Dual-High	SUB34	1	Did your strategy change over the 8 trials? [Please explain]	no--they were all right
1	Dual-High	SUB34	1	How did you decide whether to accept or reject an altitude change request? (3)	based on the size of the aircraft initially, then in cases of a moderate sized plane, I used the other information, with the assumption that the lower the turbulence and higher the fuel the greater a chance of accepting the plane for a change
1	Dual-High	SUB34	2	Did you use a rule?	Yes
1	Dual-High	SUB34	2	If Yes, I accepted an altitude	the plane was small

				request when	
1	Dual-Low	SUB35	1	How did you decide whether to accept or reject an altitude change request?	I immediately accepted planes that were S and rejected L planes.
1	Dual-Low	SUB35	1	Did your strategy change over the 8 trials? [Please explain]	NO
1	Dual-Low	SUB35	1	How did you decide whether to accept or reject an altitude change request? (3)	I immediately accepted planes that were S and XS and rejected L and XL planes. For the M planes, I looked at the turbulence level to determine whether to accept or reject. Turbulence higher than a 3, I rejected.
1	Dual-Low	SUB35	2	Did you use a rule?	Yes
1	Dual-Low	SUB35	2	If Yes, I accepted an altitude request when	When the plane was S
1	Control	SUB36	1	How did you decide whether to accept or reject an altitude change request?	I began by taking all of the factors into consideration, gas, size of plane and altitude and used my common sense to answer the questions.
1	Control	SUB36	1	Did your strategy change over the 8 trials? [Please explain]	Yes, I later found that there really was no connection between the combination of these factors with the decision to accept or reject, that the true "logic" was the size of plane, if it was S I would accept and if it was L I would reject. This strategy begot the best results.
1	Control	SUB36	1	How did you decide whether to accept or reject an altitude change request? (3)	I thought of a continuum and put XS on one end and XL on the other and used those as my determining factor, therefore XS and S were accepted and L and XL planes were rejected. It was the M sized planes that frustrated me so I used the other factors to decide, if the other numbers were high to me that leaned towards the L and XL categorization where as the lower factors with the M lead me to categorize those planes with the XS and S planes.
1	Control	SUB36	2	Did you use a rule?	Yes
1	Control	SUB36	2	If Yes, I accepted an altitude request when	the plane's size was S

3	Dual-Low	SUB37	1	How did you decide whether to accept or reject an altitude change request?	I decided on instinct.
3	Dual-Low	SUB37	1	Did your strategy change over the 8 trials? [Please explain]	Yes, at first I looked for the patterns but then I just did trial and error and went on my instinct.
3	Dual-Low	SUB37	1	How did you decide whether to accept or reject an altitude change request? (3)	I chose according to what was similar to the ones that I had remembered to be correct.
3	Dual-Low	SUB37	2	Did you use a rule?	No
3	Dual-Low	SUB37	2	If No, what did you do?	instinct
3	Dual-High	SUB38	1	How did you decide whether to accept or reject an altitude change request?	I learned to reject most small, and reject large with turbulence and fuel. At first it was guessing, but I did get some kind of pattern.
3	Dual-High	SUB38	1	Did your strategy change over the 8 trials? [Please explain]	Yes, I went from random guessing to making decisions based on the previous answers.
3	Dual-High	SUB38	1	How did you decide whether to accept or reject an altitude change request? (3)	From the trials before, I learned to accept large planes with minimal fuel and no turbulence, and to reject most small planes, I carried this over and tried to make educated guesses.
3	Dual-High	SUB38	2	Did you use a rule?	Yes
3	Dual-High	SUB38	2	If Yes, I accepted an altitude request when	I accepted large planes with minimal fuel, and reject all other large, with small I reject all except with allot of fuel and no turbulence.
3	Control	SUB39	1	How did you decide whether to accept or reject an altitude change request?	I decided to accept or reject altitude request based on the size of the plane, fuel remaining, and turbulence. High turbulence meant to me that smaller aircraft could not change altitude. Low fuel on large aircraft combined with high turbulence required an altitude change, while low turbulence and large plane allowed for a change in altitude. Small aircraft could not change altitude unless a low value of turbulence and a higher percentage of fuel were present.
3	Control	SUB39	1	Did your strategy change over the 8 trials? [Please explain]	Yes. At first I attempted to memorize which aircraft did

					what, but I got lazy and changed to something a little less memory intensive. I ended up referring to large aircraft in low turbulence as "Elrond" since Elrond was always right, and I also started to use the rule of thumb that small aircraft are always bad unless with lots of fuel and calm skies.
3	Control	SUB39	1	How did you decide whether to accept or reject an altitude change request? (3)	Shooting in the dark. I patterned the XL planes after the large planes and the XS planes after the small planes. After that it was all educated guesses based on fuel and turbulence.
3	Control	SUB39	2	Did you use a rule?	Yes
3	Control	SUB39	2	If Yes, I accepted an altitude request when	large plane and low turbulence, means the fuel load didn't matter, small planes changed altitude only in low turbulence and high fuel large planes with low fuel had to get out of high turbulence.
6	Dual-High	SUB40	1	How did you decide whether to accept or reject an altitude change request?	
6	Dual-High	SUB40	1	How did you decide whether to accept or reject an altitude change request? (3)	
6	Dual-High	SUB40	2	Did you use a rule?	No
6	Dual-High	SUB40	2	If Yes, I accepted an altitude request when	n/a
6	Dual-High	SUB40	2	If No, what did you do?	I tried to remember which ones were wrong.
6	Dual-Low	SUB41	1	How did you decide whether to accept or reject an altitude change request?	positive= 40L, 20S, or 1 negative= 20L, 40S, or 3 I rejected all double positives and double negatives I accepted the ones only with a negative and a positive
6	Dual-Low	SUB41	1	Did your strategy change over the 8 trials? [Please explain]	once I got the strategy: NO
6	Dual-Low	SUB41	1	How did you decide whether to accept or reject an altitude change request? (3)	Kind of the same manner, but when plane was less than or equal to medium and fuel was approximately matched,3, 4 and 5 turbulence became a negative which = accept

					when the planes were large or XL and fuel was approximately matched only 5 turbulence was a negative= accept too little fuel with over 3 turbulence (no matter size of plane) = reject
6	Dual-Low	SUB41	2	Did you use a rule?	Yes
6	Dual-Low	SUB41	2	If Yes, I accepted an altitude request when	when fuel and plane size matched and there was turbulence or when fuel and plane size did not match and there was no turbulence
6	Control	SUB42	1	How did you decide whether to accept or reject an altitude change request?	I tried to remember from trial to trial which combinations were accepted and which were rejected.
6	Control	SUB42	1	Did your strategy change over the 8 trials? [Please explain]	Not really, I mostly just tried to remember throughout all of the 8 trials.
6	Control	SUB42	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to see how the new combinations would relate to the old ones, and whether I had rejected the ones it seemed to match the most.
6	Control	SUB42	2	Did you use a rule?	No
6	Control	SUB42	2	If No, what did you do?	I tried to remember which combinations were accepted and which were not.
1	Dual-Low	SUB43	1	How did you decide whether to accept or reject an altitude change request?	If the turbulence was 3, I rejected the request based on the previous reinforcement I received in the first of the 8 trials. Therefore, for all of the "1" turbulences, I accepted the request.
1	Dual-Low	SUB43	1	Did your strategy change over the 8 trials? [Please explain]	No, the instructions stated that the "same rules applied" and "would not change" so I continued with my same strategy throughout the 8 trials.
1	Dual-Low	SUB43	1	How did you decide whether to accept or reject an altitude change request? (3)	I decided that any turbulence which was 3 or higher would be rejected because a 3 was rejected in the previous trials. The dilemma was determining whether or not to accept a

					request from a plane with a turbulence of "2." I decided to accept requests from planes with a turbulence of 2 for no good reason. It was strictly a personal judgment call. Of course any plane with a 0 or 1 turbulence would be accepted based on the previous trials.
1	Dual-Low	SUB43	2	Did you use a rule?	Yes
1	Dual-Low	SUB43	2	If Yes, I accepted an altitude request when	The turbulence was a 1.
1	Dual-High	SUB44	1	How did you decide whether to accept or reject an altitude change request?	1 I accepted 3 I rejected
1	Dual-High	SUB44	1	Did your strategy change over the 8 trials? [Please explain]	Yes, at first I though that size and fuel had something to do with it but after I keep getting them wrong I figured it was only turbulence that mattered
1	Dual-High	SUB44	1	How did you decide whether to accept or reject an altitude change request? (3)	When the turbulence was 0,1,2 I accepted and when it was 3 and over I rejected based on previous trails
1	Dual-High	SUB44	2	Did you use a rule?	Yes
1	Dual-High	SUB44	2	If Yes, I accepted an altitude request when	the turbulence was 1
1	Control	SUB45	1	How did you decide whether to accept or reject an altitude change request?	the fuel reserve and the size of the plane did not matter as much as the turbulence as to wither the altitude was accepted to be changed
1	Control	SUB45	1	Did your strategy change over the 8 trials? [Please explain]	no it did not
1	Control	SUB45	1	How did you decide whether to accept or reject an altitude change request? (3)	I looked at the turbulence and the size of the plane
1	Control	SUB45	2	Did you use a rule?	Yes
1	Control	SUB45	2	If Yes, I accepted an altitude request when	yes the smaller the turbulence the more likely it would be accepted for a altitude change
3	Dual-High	SUB46	1	How did you decide whether to accept or reject an altitude change request?	I tried to remember which answers I had already given were correct.
3	Dual-High	SUB46	1	Did your strategy change over the 8 trials? [Please explain]	No. I just tried to remember which answers I had given which were correct.
3	Dual-High	SUB46	1	How did you decide whether to accept or reject an altitude change request? (3)	I based my decision on the amount of turbulence and size of the plane.

3	Dual-High	SUB46	2	Did you use a rule?	No
3	Dual-High	SUB46	2	If No, what did you do?	I just tried to use my memory to decide.
3	Dual-Low	SUB47	1	How did you decide whether to accept or reject an altitude change request?	reasoning on the size of plane, fuel left and turbulence
3	Dual-Low	SUB47	1	Did your strategy change over the 8 trials? [Please explain]	no
3	Dual-Low	SUB47	1	How did you decide whether to accept or reject an altitude change request? (3)	reasoning the size of plane and fuel left
3	Dual-Low	SUB47	2	Did you use a rule?	Yes
3	Dual-Low	SUB47	2	If Yes, I accepted an altitude request when	a small plane had a high turbulent and not much fuel left
3	Control	SUB48	1	How did you decide whether to accept or reject an altitude change request?	I decided to accept an altitude change if the aircraft had sufficient amount of fuel so that it would steady the plane as it climbed in altitude also id there was a lot of turbulence the aircraft had to be carrying 40 gallons if it were either large or small. I fit was small it was accepted for change in altitude 40 S 3, only. If it were a large plane it would be accepted if it were 40 L 1, 40 L 3, and 20 L 1 Rejections: Small would be rejected if it were 20 S 1, 20 S 3, and 40 S 1
3	Control	SUB48	1	Did your strategy change over the 8 trials? [Please explain]	at first I was unsure of the strategy but I figured it out after the third trial. Any mistakes that were made were because I was careless. My strategy remained the same through out the entire experiment.
3	Control	SUB48	1	How did you decide whether to accept or reject an altitude change request? (3)	I used the same strategy but it was difficult because I was unsure if the Small or Large plane could handle 30 gallons of fuel and a 3 with turbulence, I would have known if there was a smiley face.
3	Control	SUB48	2	Did you use a rule?	Yes
3	Control	SUB48	2	If Yes, I accepted an altitude request when	Small: 40 S 3 I accepted when a small plane had enough fuel to stabilize it in the turbulence as it

					elevated
6	Dual-Low	SUB49	1	How did you decide whether to accept or reject an altitude change request?	Memorize two of the combinations and figured the rest out. I memorized 40 L 1= rejected 20 S 1 = rejected Therefore 40 L 3 = accept 20 S 3 = accept And then the opposite 40 S 3 reject 40 S 1 accept 20 L 3 reject 20 L 1 accept
6	Dual-Low	SUB49	1	Did your strategy change over the 8 trials? [Please explain]	No
6	Dual-Low	SUB49	1	How did you decide whether to accept or reject an altitude change request? (3)	In the beginning, a Large plain with 40% fuel and little turbulence got a rejected request and a great amount of turbulence requests was accepted. For the small plains, 20% of fuel and little turbulence was accepted and vice versa. I thought that in big planes, turbulence was affecting the decision, meanwhile in little planes, the level off fuel was the determining factor.
6	Dual-Low	SUB49	2	Did you use a rule?	Yes
6	Dual-Low	SUB49	2	If Yes, I accepted an altitude request when	Large plains had greater amount of turbulence and when small planes had greater amount of fuel.
6	Dual-High	SUB50	1	How did you decide whether to accept or reject an altitude change request?	I accepted the planes that had little fuel and that were encountering turbulence. The size of the plane factored into my decision based on the smaller the more likely I would grant when factored in with the other data given.
6	Dual-High	SUB50	1	Did your strategy change over the 8 trials? [Please explain]	No, my strategy really didn't change over the eight trials
6	Dual-High	SUB50	1	How did you decide whether to accept or reject an altitude change request? (3)	I accepted the planes on the same premise as I listed above.
6	Dual-High	SUB50	2	Did you use a rule?	Yes
6	Dual-High	SUB50	2	If Yes, I accepted an altitude request when	I thought that if the fuel was lower and the turbulence was high I would grant the change. Plus, the size of the plane would factor into it.
6	Control	SUB51	1	How did you decide whether to accept or reject an altitude	partially by locale, also weighed in whether they

				change request?	had a lot of fuel and their size. A large plane with low fuel seemed to not be able to make the change. Small planes with high fuel seemed to be able to make the transition.
6	Control	SUB51	1	Did your strategy change over the 8 trials? [Please explain]	A few times. The original method used did not use the information provided to help make the decision, therefore I had to figure out the pertinence of that info.
6	Control	SUB51	1	How did you decide whether to accept or reject an altitude change request? (3)	A large plane with enough fuel was able to withstand the turbulence, while the small planes needed the fuel to transition them through the turbulence. The extra variables thrown in I wasn't sure how to deal with so I just tried to remain consistent with my first choices.
6	Control	SUB51	2	Did you use a rule?	No
6	Control	SUB51	2	If No, what did you do?	I changed the method used several.times to see if there may have been another strategy I wasn't addressing.
1	Dual-High	SUB52	1	How did you decide whether to accept or reject an altitude change request?	I would accept if it was a 1 but reject if it was a 3.
1	Dual-High	SUB52	1	Did your strategy change over the 8 trials? [Please explain]	Yes, at first I was looking at the 40 or 20, then to the L or S, then I was able to figure out how to accept or reject.
1	Dual-High	SUB52	1	How did you decide whether to accept or reject an altitude change request? (3)	If it was a 3 or 4 I rejected it, but if it was a 0,1,2 I accepted the change simply because they were low numbers.
1	Dual-High	SUB52	2	Did you use a rule?	Yes
1	Dual-High	SUB52	2	If Yes, I accepted an altitude request when	The number was low
1	Dual-Low	SUB53	1	How did you decide whether to accept or reject an altitude change request?	Trying to correlate the variable(plane size, fuel, and turbulence)
1	Dual-Low	SUB53	1	Did your strategy change over the 8 trials? [Please explain]	If I had the smiley face I would do the same thing in the next request, otherwise I would try to change my strategy.
1	Dual-Low	SUB53	1	How did you decide whether	Based on the previous

				to accept or reject an altitude change request? (3)	trials.
1	Dual-Low	SUB53	2	Did you use a rule?	Yes
1	Dual-Low	SUB53	2	If Yes, I accepted an altitude request when	the turbulence was 1
1	Control	SUB54	1	How did you decide whether to accept or reject an altitude change request?	I decided to accept or reject an altitude change request according to the combination of the provided variables. For instance, if there was low to no turbulence and low gas, I would reject it. Or, if there was high turbulence and high gas I would accept it. The size of the plane meant approx. how much gas it would need, so the lower the gas level, the less like I would be to accept a request. Also, the larger the plane, the heavier it had to be, so a large plane would probably be able to handle more turbulence than a smaller one. This is what my decisions were based upon.
1	Control	SUB54	1	Did your strategy change over the 8 trials? [Please explain]	No, my strategy did not change over the 8 trials; the only thing that may have changed was that I gained the insight that my strategy was accurate, even when I guessed incorrectly.
1	Control	SUB54	1	How did you decide whether to accept or reject an altitude change request? (3)	I decided to accept or reject an altitude change request in the same manner that I did in the other eight trials, this time there was just more variable information and less time for reaction.
1	Control	SUB54	2	Did you use a rule?	Yes
1	Control	SUB54	2	If Yes, I accepted an altitude request when	there was greater amounts of gas and high altitude, or medium gas, small plane, and high altitude
3	Dual-Low	SUB55	1	How did you decide whether to accept or reject an altitude change request?	I decided based on looking at the three values that they told me to look at, and tried to make a decision from there. I mostly looked at the amount of fuel. It seemed as though the more fuel there was left, the better

					chance of getting an altitude change. I did not really focus too much on the turbulents and the size of the plane, because it did not seem as though that had as much of an effect as did the amount of fuel left.
3	Dual-Low	SUB55	1	Did your strategy change over the 8 trials? [Please explain]	No I kept the same strategy throughout the 8 trials. If I switched, it may have became more confusing.
3	Dual-Low	SUB55	1	How did you decide whether to accept or reject an altitude change request? (3)	Some of them I already knew from the 8 trials beforehand. The other ones I decided based on the values that I was given, again I focused mostly on the amount of fuel left than any other value that I was given.
3	Dual-Low	SUB55	2	Did you use a rule?	Yes
3	Dual-Low	SUB55	2	If Yes, I accepted an altitude request when	I accepted an altitude request when the amount of fuel was over 40, I had although accepted one 20 which was 20 S 1. So as a rule, after most of the trials, I had remembered which ones were accepted, and which ones were rejected.
3	Dual-High	SUB56	1	How did you decide whether to accept or reject an altitude change request?	I took into account the percentage fuel, size of the plane, and the turbulence as whole. I put all three pieces of information together and remembered which ones where the correct ones and which were not.
3	Dual-High	SUB56	1	Did your strategy change over the 8 trials? [Please explain]	My strategy basically remained constant. The magenta planes remained with the same three components that I had observed throughout the experiment.
3	Dual-High	SUB56	1	How did you decide whether to accept or reject an altitude change request? (3)	The fuel remaining, size of plane, and turbulence were all compared with one another. I focused on the fuel remaining with the turbulence and decided if they worked together. Such as, if the fuel was an even

					number and the turbulence was also, then I would accept it.
3	Dual-High	SUB56	2	Did you use a rule?	Yes
3	Dual-High	SUB56	2	If Yes, I accepted an altitude request when	a certain fuel amount was paired with the size of a plane and turbulence. If such component did not appear, than it would be rejected.
3	Control	SUB57	1	How did you decide whether to accept or reject an altitude change request?	Turbulence. I interpreted the amount of turbulence that had occurred as where the plane was altitude wise. This helped me picture the planes and apply the other aspects.
3	Control	SUB57	1	Did your strategy change over the 8 trials? [Please explain]	No, but I did learn from my mistakes.
3	Control	SUB57	1	How did you decide whether to accept or reject an altitude change request? (3)	The larger the property, the more I connected them with the previous examples. The medium, or average, variables were the most confusing as I had no basis of comparison.
3	Control	SUB57	2	Did you use a rule?	Yes
3	Control	SUB57	2	If Yes, I accepted an altitude request when	The plane was small and had little fuel, I wouldn't let accept. When the plane was small and had enough fuel to move out of high turbulence, then I accepted. Larger planes could move whenever, except when they had low fuel and high turbulence.
6	Dual-High	SUB58	1	How did you decide whether to accept or reject an altitude change request?	large planes with Low turbulence, 40 fuel (accept) Small planes with low turbulence and 40 fuel (accept) Large planes with low fuel, (reject) Small planes with high turbulence and low fuel (reject) Large plans with high turbulence (accept) Small planes with high turbulence and High fuel (accept) Other times I accepted every other plane.
6	Dual-High	SUB58	1	Did your strategy change over the 8 trials? [Please explain]	I thought at first that all planes with 40 fuel were to be rejected, and all planes with 20 fuel to be accepted. It was hard not to think

					logically about whether or not the planes should be allowed to increase their altitude. Smaller planes with less fuel and a high turbulence, to me shouldn't be allowed an increase on their altitude.
6	Dual-High	SUB58	1	How did you decide whether to accept or reject an altitude change request? (3)	I assumed that it was the same idea, just on a larger scale. Higher fuel, larger plane, and altitude. Smaller planes I thought shouldn't be increasing with a high turbulence factor.
6	Dual-High	SUB58	2	Did you use a rule?	Yes
6	Dual-High	SUB58	2	If Yes, I accepted an altitude request when	at first I tried accepting every other one. When that wasn't working, I tried accepting all 20 fuel, and rejecting all 40 fuel. That was not working. So I tried using common sense, rejecting small, high turbulence planes with low fuel. Large low fuel planes with high turbulence (accept) Large planes with low fuel with low turbulence (reject) All planes with 40% fuel I accepted. I also tried rejecting all Large and accepting all small, and vice versa.
6	Dual-Low	SUB59	1	How did you decide whether to accept or reject an altitude change request?	I accepted all of the ones with the 20% fuel remaining if the size and the turbulence was the same as in it being either small and 1 or large and 3. If it was small and 3 which is opposite I rejected it. The 40% fuel remaining I accepted it if it was opposite as in the turbulence being a 3 and the size being s, I rejected it if it was the same as in the size being s and the turbulence being a 1.
6	Dual-Low	SUB59	1	Did your strategy change over the 8 trials? [Please explain]	No
6	Dual-Low	SUB59	1	How did you decide whether to accept or reject an altitude change request? (3)	The first 4 I just did whatever. The next couple of ones I did 10% accept if

					it was opposite within 2 intervals; 20% the same within 2 intervals; 30% opposite within intervals; 40% same within 2 intervals and 50% opposite within 2 intervals. I then realized that the 40% should be accepted if it was opposite like before so I changed it to accepting 10 and 20 percent within 2 intervals. I accepted 30 and 40 percent if they were opposite within 2 intervals and 50% I accepted it if was the same within 20 intervals. example: Accept 10 XL 1
6	Dual-Low	SUB59	2	Did you use a rule?	Yes
6	Dual-Low	SUB59	2	If Yes, I accepted an altitude request when	For 20% if the size and turbulence were the same i.e., if they were both small. I rejected the 40% if they were opposite.
6	Control	SUB60	1	How did you decide whether to accept or reject an altitude change request?	the turbulence levels/ how close they were to the intersection with other planes.
6	Control	SUB60	1	Did your strategy change over the 8 trials? [Please explain]	At first I tried to figure out different formulas for the right and wrong answer, nothing except the distance between the planes seemed to make any sense. Yet, I still got the wrong answers! So by the end I tried to use reasoning, and that to failed me.
6	Control	SUB60	1	How did you decide whether to accept or reject an altitude change request? (3)	Distance apart from the planes, and the turbulence it may cause for ensuing planes.
6	Control	SUB60	2	Did you use a rule?	Yes
6	Control	SUB60	2	If Yes, I accepted an altitude request when	planes were in the clear of other planes and would not cause excessive turbulence for surrounding planes
1	Dual-Low	SUB61	1	How did you decide whether to accept or reject an altitude change request?	accept large planes, reject small planes. After half way through the first test, I used size to determine whether to accept or reject, fuel and turbulence I ignored.

1	Dual-Low	SUB61	1	Did your strategy change over the 8 trials? [Please explain]	during the first trial I thought there was a relationship between fuel remaining and size of plane, but after a few wrong answers I found that all large planes should be accepted and small rejected.
1	Dual-Low	SUB61	1	How did you decide whether to accept or reject an altitude change request? (3)	L and XL I accepted following the first 8 tests experience S and XS I rejected following the first 8 tests experience M I was not sure about, but accepted all of them for consistency
1	Dual-Low	SUB61	2	Did you use a rule?	Yes
1	Dual-Low	SUB61	2	If Yes, I accepted an altitude request when	Large plane Ignored other two factors
1	Dual-High	SUB62	1	How did you decide whether to accept or reject an altitude change request?	I rejected the small and accepted the large
1	Dual-High	SUB62	1	Did your strategy change over the 8 trials? [Please explain]	no
1	Dual-High	SUB62	1	How did you decide whether to accept or reject an altitude change request? (3)	based on the size of the plane
1	Dual-High	SUB62	2	Did you use a rule?	Yes
1	Dual-High	SUB62	2	If Yes, I accepted an altitude request when	the plane was larger
1	Control	SUB63	1	How did you decide whether to accept or reject an altitude change request?	If it was L I accepted. if it was S I rejected
1	Control	SUB63	1	Did your strategy change over the 8 trials? [Please explain]	It took me until the 3rd or 4th trial to figure out my strategy to accept or reject.
1	Control	SUB63	1	How did you decide whether to accept or reject an altitude change request? (3)	If it was M, L, or XL I accepted. If it was XS or S I rejected.
1	Control	SUB63	2	Did you use a rule?	Yes
1	Control	SUB63	2	If Yes, I accepted an altitude request when	accepted when Large
3	Dual-High	SUB64	1	How did you decide whether to accept or reject an altitude change request?	The basis of my acceptance was between the how high the level of turbulence was and how much fuel was left, more so than the size of the plane. The higher the turbulence and lower the fuel level led me to accept their request. For example a 20 S 3 was always accepted, and was always correct.

3	Dual-High	SUB64	1	Did your strategy change over the 8 trials? [Please explain]	I stayed consistent with my strategy for every trial. Simply, because it would be easier to work through. I felt my strategy was more commonsense; a small plane experiencing heavy turbulence and light on fuel would definitely need to make some adjustments.
3	Dual-High	SUB64	1	How did you decide whether to accept or reject an altitude change request? (3)	I stayed with the basis of my earlier trials, keeping the idea that more turbulence and less fuel levels would constitute for an adjustment in altitude. Hopefully, making the plane easier to handle and in turn not use as much fuel.
3	Dual-High	SUB64	2	Did you use a rule?	Yes
3	Dual-High	SUB64	2	If Yes, I accepted an altitude request when	the planes were light on fuel and heavy in turbulence, I did not take into account the size of the plane, because I figured any plane experiencing heavy turbulence and light fuel levels would need to make adjustments.
3	Dual-Low	SUB65	1	How did you decide whether to accept or reject an altitude change request?	Followed the pattern I found in the earlier trials, since I figured out what the pattern was I just had to go by that.
3	Dual-Low	SUB65	1	Did your strategy change over the 8 trials? [Please explain]	Yes, at first it was trial and error, just trying a pattern and then another one until I found what the pattern was.
3	Dual-Low	SUB65	1	How did you decide whether to accept or reject an altitude change request? (3)	Tried to fit them in as best I could with the pattern from the earlier trials.
3	Dual-Low	SUB65	2	Did you use a rule?	Yes
3	Dual-Low	SUB65	2	If Yes, I accepted an altitude request when	It was at 20, L or S and had 3, or was at 20, S with 1. Or was at 40, L, 3.
3	Control	SUB66	1	How did you decide whether to accept or reject an altitude change request?	I memorized the answers. i.e., 20 S 1, 40 L 3, 20S/L3 I accepted, and rejected everything else.
3	Control	SUB66	1	Did your strategy change over the 8 trials? [Please explain]	No
3	Control	SUB66	1	How did you decide whether to accept or reject an altitude change request? (3)	Tried to match the numbers with the ones from the first 8 trials
3	Control	SUB66	2	Did you use a rule?	Yes

3	Control	SUB66	2	If Yes, I accepted an altitude request when	The rule that I previously stated in question #1.
6	Dual-Low	SUB67	1	How did you decide whether to accept or reject an altitude change request?	the higher the turbulence determined whether or not to accept or reject. If the turbulence was high, I would accept an altitude change. Also if the gas was low I would accept the turbulence change.
6	Dual-Low	SUB67	1	Did your strategy change over the 8 trials? [Please explain]	Towards the end it changed. I'm still not too sure if that was the right strategy.
6	Dual-Low	SUB67	1	How did you decide whether to accept or reject an altitude change request? (3)	I took into consideration the size of the plane; if the plane was large it could go through more turbulence than the small ones. Also, the amount of gasoline left in the plane was also important. If there was not too much fuel left, the altitudes needed to be changed.
6	Dual-Low	SUB67	2	Did you use a rule?	No
6	Dual-Low	SUB67	2	If No, what did you do?	I compared the size, turbulence and gas. The larger planes need more gas, and could undergo more turbulence.
6	Dual-High	SUB68	1	How did you decide whether to accept or reject an altitude change request?	I compared how much turbulence there was to the size of the plane. Relatively, if the plane did not have much fuel, then sometimes I would decide to decline altitude request.
6	Dual-High	SUB68	1	Did your strategy change over the 8 trials? [Please explain]	Yes. Certain situations that I thought were correct based on the strategy were actually wrong. If this happened repeatedly, I would adjust to it. So, I memorized the specific details and gave the same indication for those planes.
6	Dual-High	SUB68	1	How did you decide whether to accept or reject an altitude change request? (3)	If there was a great deal of turbulence, then I would reject most planes in general. With moderate turbulence, I would online permit medium or large planes. Also, if a plane had

					very low fuel, I would decline--especially if it was of small size.
6	Dual-High	SUB68	2	Did you use a rule?	Yes
6	Dual-High	SUB68	2	If Yes, I accepted an altitude request when	If the plane was large and the turbulence was low.
6	Control	SUB69	1	How did you decide whether to accept or reject an altitude change request?	At first I looked at fuelage. Ex. 40 L 1, I would accept.
6	Control	SUB69	1	Did your strategy change over the 8 trials? [Please explain]	I started looking at fuelage, then turbulence.
6	Control	SUB69	1	How did you decide whether to accept or reject an altitude change request? (3)	I kept the same strategy from ques. #2 with a slight focus on plane size.
6	Control	SUB69	2	Did you use a rule?	Yes
6	Control	SUB69	2	If Yes, I accepted an altitude request when	I focused mostly on fuel and turbulence.
1	Dual-High	SUB70	1	How did you decide whether to accept or reject an altitude change request?	Sometimes I based my decision on the amount of fuel, other times I just guessed
1	Dual-High	SUB70	1	Did your strategy change over the 8 trials? [Please explain]	Yes, When one thing didn't work I tried another thing.
1	Dual-High	SUB70	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to accept more than reject in order not to lose points.
1	Dual-High	SUB70	2	Did you use a rule?	Yes
1	Dual-High	SUB70	2	If Yes, I accepted an altitude request when	I accepted an altitude request not lose points.
1	Dual-Low	SUB71	1	How did you decide whether to accept or reject an altitude change request?	If it was a small plane, I accepted the request. If it was a large plane, I rejected the request.
1	Dual-Low	SUB71	1	Did your strategy change over the 8 trials? [Please explain]	Yes. At first, I was lost as to how to accept or reject a plane's altitude change. But by the 6th or 7th trial, I realized it was the size of the plane.
1	Dual-Low	SUB71	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to combine what I knew about size from the previous trials to the planes in this trial. I also took into account the turbulence and the size. If it was XS, I treated it like I would treat a small plane in the previous trial.
1	Dual-Low	SUB71	2	Did you use a rule?	Yes
1	Dual-Low	SUB71	2	If Yes, I accepted an altitude request when	The plane was small.
1	Control	SUB72	1	How did you decide whether to accept or reject an altitude change request?	if it was small I accepted if it was large I rejected

1	Control	SUB72	1	Did your strategy change over the 8 trials? [Please explain]	no
1	Control	SUB72	1	How did you decide whether to accept or reject an altitude change request? (3)	if it was XS, S, or M I accepted if it was L or XL I rejected
1	Control	SUB72	2	Did you use a rule?	Yes
1	Control	SUB72	2	If Yes, I accepted an altitude request when	it was small
3	Dual-Low	SUB73	1	How did you decide whether to accept or reject an altitude change request?	I accepted altitude changes for 40 S' with 1 or 3. I rejected altitude changes for 40 L's with 1 or 3. For all 20's with three's I accepted and for all 20's with ones I rejected.
3	Dual-Low	SUB73	1	Did your strategy change over the 8 trials? [Please explain]	Yes I came up with several strategies and through trial and error I found the most productive one and stuck with that one.
3	Dual-Low	SUB73	1	How did you decide whether to accept or reject an altitude change request? (3)	I accepted all planes with XL, or 0's. I continued to accept and reject by the strategy I had used successfully in the 8 trials before. I rejected all the other combos.
3	Dual-Low	SUB73	2	Did you use a rule?	Yes
3	Dual-Low	SUB73	2	If Yes, I accepted an altitude request when	when the fuel was 40 and the plane was small or when the fuel was 20 and the turbulence was three.
3	Dual-High	SUB74	1	How did you decide whether to accept or reject an altitude change request?	I memorized those plane characteristics that resulted in an indication of an incorrect response for acceptance and rejection of altitude requests. I also attempted to apply those responses that were correct to decide what response to provide for new combinations of plane characteristics.
3	Dual-High	SUB74	1	Did your strategy change over the 8 trials? [Please explain]	No
3	Dual-High	SUB74	1	How did you decide whether to accept or reject an altitude change request? (3)	I attempted to use the same principles for the original eight trials (i.e., large planes with low fuel and high turbulence should be allowed to change altitudes). However for the most part my decisions for those plane characteristics

					that were new to the presented situation were arbitrary. This was due mainly to a lack of feedback regarding correct and incorrect decisions as the trial progressed.
3	Dual-High	SUB74	2	Did you use a rule?	Yes
3	Dual-High	SUB74	2	If Yes, I accepted an altitude request when	40 percent of fuel was remaining and the plane was small; also, when the plane was small with 20 percent of fuel remaining but with a high turbulence level.
3	Control	SUB75	1	How did you decide whether to accept or reject an altitude change request?	by determining what the size of the plane was and the turbulence
3	Control	SUB75	1	Did your strategy change over the 8 trials? [Please explain]	yes, I had no clue what to look for at first.
3	Control	SUB75	1	How did you decide whether to accept or reject an altitude change request? (3)	The same way I did it for the first exercise
3	Control	SUB75	2	Did you use a rule?	Yes
3	Control	SUB75	2	If Yes, I accepted an altitude request when	I saw the size of the plane was large enough to withstand the turbulence that was placed upon it.
6	Dual-High	SUB76	1	How did you decide whether to accept or reject an altitude change request?	Based on the amount of turbulence and how much fuel was left...I also remembered a few combinations (i.e., 40 L 1)
6	Dual-High	SUB76	1	Did your strategy change over the 8 trials? [Please explain]	Not really, I wasn't able to pick up a pattern that well, so I just remembered what I could and guessed on the rest
6	Dual-High	SUB76	1	How did you decide whether to accept or reject an altitude change request? (3)	I mostly guessed...again I remembered a few combinations from the previous trials (40 L 1) ...usually if the turbulence was high, I would reject the request, but later changed my strategy and accepted requests with high levels of turbulence and high fuel.
6	Dual-High	SUB76	2	Did you use a rule?	No
6	Dual-High	SUB76	2	If No, what did you do?	I just remembered one or two that worked and went with my gut feelings on certain ones. Mostly if they had enough fuel (30% or higher) I would accept the

					request for an altitude change.
6	Dual-Low	SUB77	1	How did you decide whether to accept or reject an altitude change request?	I used % of the fuel and direction of the airplane as cues to decide whether to accept or reject an altitude change request.
6	Dual-Low	SUB77	1	Did your strategy change over the 8 trials? [Please explain]	Yes. Sometimes I changed my strategy based on the result.
6	Dual-Low	SUB77	1	How did you decide whether to accept or reject an altitude change request? (3)	I used % of the fuel and the size of the airplanes as cues to make my decisions, but I also used my wild guess.
6	Dual-Low	SUB77	2	Did you use a rule?	Yes
6	Dual-Low	SUB77	2	If Yes, I accepted an altitude request when	the percentage of the fuel was high.
6	Control	SUB78	1	How did you decide whether to accept or reject an altitude change request?	At first by memorization. Then I was convinced that there wasn't a pattern for a while. Then I tried memorization again.
6	Control	SUB78	1	Did your strategy change over the 8 trials? [Please explain]	It changed from time to time when I wasn't sure that there was a pattern. Sometimes I guessed without paying attention out of boredom, other times I did try to memorize numbers.
6	Control	SUB78	1	How did you decide whether to accept or reject an altitude change request? (3)	It was based entirely on the amount of turbulence in comparison with the size of the plane and how much fuel it had. The more need based the altitude change was, the more likely I was to accept it.
6	Control	SUB78	2	Did you use a rule?	No
6	Control	SUB78	2	If No, what did you do?	Simple memorization of answers.
1	Dual-Low	SUB79	1	How did you decide whether to accept or reject an altitude change request?	If the turbulence was 3 I accepted the request for change of altitude if it was 1 I rejected the request.
1	Dual-Low	SUB79	1	Did your strategy change over the 8 trials? [Please explain]	No
1	Dual-Low	SUB79	1	How did you decide whether to accept or reject an altitude change request? (3)	If the turbulence was 2 or greater I accepted the request for a change in altitude if it was 1 or 0 I denied the request for change in altitude.
1	Dual-Low	SUB79	2	Did you use a rule?	Yes

1	Dual-Low	SUB79	2	If Yes, I accepted an altitude request when	I accepted the request when the turbulence was 3
1	Dual-High	SUB80	1	How did you decide whether to accept or reject an altitude change request?	Depends on how close the airplane was
1	Dual-High	SUB80	1	Did your strategy change over the 8 trials? [Please explain]	yes
1	Dual-High	SUB80	1	How did you decide whether to accept or reject an altitude change request? (3)	Commonsense ...It depends how far the airplane was
1	Dual-High	SUB80	2	Did you use a rule?	No
1	Dual-High	SUB80	2	If No, what did you do?	Randomly reject it
1	Control	SUB81	1	How did you decide whether to accept or reject an altitude change request?	If turbulence was high, 3, the request was accepted; 1 for turbulence was rejected.
1	Control	SUB81	1	Did your strategy change over the 8 trials? [Please explain]	Only at the beginning of the first trial until I figured out the pattern.
1	Control	SUB81	1	How did you decide whether to accept or reject an altitude change request? (3)	Since size and fuel did not matter in the first 8 trials, I followed my turbulence pattern from before; 1 was always rejected, 3 was always accepted. By that same logic, 0 was always rejected, and 4 always accepted as well. A 2 for turbulence was the most difficult, since it fell between my rules, so I based my decision on the size of the plane. A smaller plane would not be able to handle turbulence as well as a larger plane, so the small and extra small planes were given acceptance.
1	Control	SUB81	2	Did you use a rule?	Yes
1	Control	SUB81	2	If Yes, I accepted an altitude request when	Turbulence was 3.
3	Dual-High	SUB82	1	How did you decide whether to accept or reject an altitude change request?	I accepted altitude change requests under the following conditions: a. 20% of fuel remaining, small plane, level 1 turbulence; or b. 20% fuel remaining, large plane, level 1 turbulence; or c. either 20 or 40% fuel remaining, large plane, level 3 turbulence
3	Dual-High	SUB82	1	Did your strategy change over the 8 trials? [Please explain]	The first trial I guessed and realized that conditions A or

					C (from question 1) must be met in order to accept the request. In the third trial I realized that condition b (from question 1) was another condition under which the request could be accepted.
3	Dual-High	SUB82	1	How did you decide whether to accept or reject an altitude change request? (3)	I grouped the small planes (XS and S) together and if they had a small percent of fuel remaining (20% or below) and a low level of turbulence (1 or below) I accepted the altitude change request. I grouped the large planes together (L and XL) and if they had a small percent of fuel remaining (20% or below) and a low level (1 or below) of turbulence, I accepted the request. In addition, regardless of the percent of fuel remaining, I accepted the request if they had a high level of turbulence (level 3 or above). It was difficult to decide how to categorize the medium planes so I accepted the request if they had a medium percent of fuel remaining (20 or 30%) and if they were at a medium level of turbulence (2). In the beginning of the task, I had not clearly determined how I was going to categorize the different planes, so I may not have followed the above strategy for the first couple of planes.
3	Dual-High	SUB82	2	Did you use a rule?	Yes
3	Dual-High	SUB82	2	If Yes, I accepted an altitude request when	I accepted an altitude request under the following conditions: a. 20% fuel remaining, small plane, level 1 turbulence; b. 20% fuel, large plane, level 1 turbulence; and c. 20 or 40% fuel, large plane, level 3 turbulence.
3	Dual-Low	SUB83	1	How did you decide whether	I memorized the number

				to accept or reject an altitude change request?	and letter patterns. For example 20 L 3 were always accepted and 20 S 3 were always rejected.
3	Dual-Low	SUB83	1	Did your strategy change over the 8 trials? [Please explain]	Yes. First I thought you had to change the altitudes when two planes were about to crash. Then I thought it dealt with the N/S and E/W directions. finally I just settled on memorizing the different number groups.
3	Dual-Low	SUB83	1	How did you decide whether to accept or reject an altitude change request? (3)	I tried to use the ratios of the answers I knew were correct and applied them to the new problems.
3	Dual-Low	SUB83	2	Did you use a rule?	No
3	Dual-Low	SUB83	2	If No, what did you do?	I tried to use the ratios of the number patterns I already recognized. On the moving trials I just tried to recognize the patterns that I was familiar with.
3	Control	SUB84	1	How did you decide whether to accept or reject an altitude change request?	The first couple of trials were guess and check, but over the next couple of trials I started to see a pattern.
3	Control	SUB84	1	Did your strategy change over the 8 trials? [Please explain]	Yes, once I started to see a pattern I changed the way I responded accordingly. I rejected every plane with 40% fuel and turbulence = 1, regardless of size and I also rejected any small plane with a turbulence =3. I accepted any large plane except for the case noted above.
3	Control	SUB84	1	How did you decide whether to accept or reject an altitude change request? (3)	I applied the same pattern to the XS planes as I did to the S planes and the same pattern to the XL planes as I did to the L planes. On the medium planes, it was kind of guess and check.
3	Control	SUB84	2	Did you use a rule?	Yes
3	Control	SUB84	2	If Yes, I accepted an altitude request when	It was a large plane with any turbulence or a small plane with a turbulence = 1
6	Dual-Low	SUB85	1	How did you decide whether to accept or reject an altitude change request?	I gave up reasoning; my answers became solely trained response as learned through the sound

					of a correct answer as opposed to the sound of the incorrect answer.
6	Dual-Low	SUB85	1	Did your strategy change over the 8 trials? [Please explain]	According to my scores they did. There was a point where I had a trained response to familiar characteristic of planes requesting an altitude change. I gave up reasoning and devised a way to remember which planes to grant requests to and which not according to my memory.
6	Dual-Low	SUB85	1	How did you decide whether to accept or reject an altitude change request? (3)	I feel I had a better grasp of what was necessary for a plane to be granted its altitude request. Obviously size, fuel percentage and turbulence are factors. The sound of a correct answer certainly helped in the others, I don't feel I understand completely what constitutes the granting of an altitude change. From what I gathered during the previous 8 trials, I felt in order for a plane to be granted a request the size and turbulence was more of a factor than the percent of fuel remaining. The higher the turbulence the more important I felt a plane to be granted its altitude request.
6	Dual-Low	SUB85	2	Did you use a rule?	No
6	Dual-Low	SUB85	2	If No, what did you do?	I was able to keep in mind, and recall which characteristic of a plane with a request was acceptable and which was not. And I devised an understanding with juxtaposing these various characteristics with each other. For instance 20 S 1 would always be no or wrong and 20 L 1 would always be correct...I believe.
6	Dual-High	SUB86	1	How did you decide whether	After the first couple of

				to accept or reject an altitude change request?	trials, I noticed a pattern. For example, I knew that if it was 20 S 3 it had to be true and if it was 40 S 3 it had to be false. I just assumed the opposite: If 20 S 3 was true, then 40 S 3 had to be false and so on.
6	Dual-High	SUB86	1	Did your strategy change over the 8 trials? [Please explain]	My strategy proved to be correct as I scored perfectly on the last couple of trials. After a while, everything became second nature as I had adapted to my strategy. For example, whenever I saw 20 L 1, I knew it had to be an accepted altitude.
6	Dual-High	SUB86	1	How did you decide whether to accept or reject an altitude change request? (3)	It was difficult to interpret the right answer. Aside from knowing the answers from the previous trials, the answers to the last trial were based on whim.
6	Dual-High	SUB86	2	Did you use a rule?	Yes
6	Dual-High	SUB86	2	If Yes, I accepted an altitude request when	If I saw 40 S 1, I assumed it would be an accepted altitude because 40 S 3 was a rejected one. I used this rule for all the other altitudes.
6	Control	SUB87	1	How did you decide whether to accept or reject an altitude change request?	At first my strategy was more complicated than necessary. I looked at the direction of the plane, and chose reject for each, until I discovered which was correct in each direction. This, however, cost me many points, until I discovered it was merely 8 matches that needed to be memorized, versus the 32 I had originally thought taking into account the direction of the plane. Being only a possible 8 matches, it was easy to discover the 4 correct answers- 20S3, 40S1, 20L1, 40L3.
6	Control	SUB87	1	Did your strategy change over the 8 trials? [Please explain]	It changed from trial 2 to trial 3. I thought I had it figured it out in trial 1, thinking that a 3 turbulence

					was incorrect in most cases, and a 1 turbulence was correct. However, then I changed my strategy to solely rejecting planes until I realized the correct answers. It seemed in the beginning that more planes were rejected than accepted, and that I was wasting points guessing the correct ones. My strategy finally changed when I realized the simplicity of the problem in trial 4. I had been attempting a more complicated approach until this point. After the realization, the trials became repetitive and tedious.
6	Control	SUB87	1	How did you decide whether to accept or reject an altitude change request? (3)	I based my decisions on the results of the first 8 trials. I stuck with the general idea of large planes having both high-high or low-low characteristics as they related to fuel and turbulence. The small planes were a little trickier in that low fuel meant high turbulence and vice- versa. Thus, it was difficult to decide what to do with the medium planes. I did not know whether to go with the characteristics of the small or large planes.
6	Control	SUB87	2	Did you use a rule?	Yes
6	Control	SUB87	2	If Yes, I accepted an altitude request when	it was only 20S3, 40S1, 20L1, 40L3.
1	Dual-High	SUB88	1	How did you decide whether to accept or reject an altitude change request?	I accepted requests only from small planes because initially when I accepted a request from a large plane, I ended up being wrong.
1	Dual-High	SUB88	1	Did your strategy change over the 8 trials? [Please explain]	My strategy didn't change. Once I knew to accept only small planes for an altitude change, and that this was my first priority, I didn't change my strategy.
1	Dual-High	SUB88	1	How did you decide whether to accept or reject an altitude change request? (3)	I used the same strategy as last time. I accepted both types of small planes, and

					rejected both types of large planes. I wasn't exactly sure what to do with the medium planes because the size was all that I paid attention to previously, so the fuel and turbulence weren't able to help me at all in reference to the medium planes.
1	Dual-High	SUB88	2	Did you use a rule?	Yes
1	Dual-High	SUB88	2	If Yes, I accepted an altitude request when	The plane was small.
1	Dual-Low	SUB89	1	How did you decide whether to accept or reject an altitude change request?	By the size of the plane. If it was a small plane, I accepted. If it was a large plane, I rejected the request.
1	Dual-Low	SUB89	1	Did your strategy change over the 8 trials? [Please explain]	No. Because I did not receive any negative feedback for maintaining the same strategy, I did not change it through the 8 trials.
1	Dual-Low	SUB89	1	How did you decide whether to accept or reject an altitude change request? (3)	By size again, and the other factors when it came to medium size planes. If the plane was an S or XS, I accepted the request. If it was an L or XL, I rejected the request. If it was a medium, with a low fuel percentage, I accepted the request. If it had a high fuel percentage, I rejected it.
1	Dual-Low	SUB89	2	Did you use a rule?	Yes
1	Dual-Low	SUB89	2	If Yes, I accepted an altitude request when	The plane size was small.
1	Control	SUB90	1	How did you decide whether to accept or reject an altitude change request?	by looking at the % of fuel left and the amount of turbulence
1	Control	SUB90	1	Did your strategy change over the 8 trials? [Please explain]	somewhat at times I looked at the size and fuel percentage
1	Control	SUB90	1	How did you decide whether to accept or reject an altitude change request? (3)	by looking at the size and fuel
1	Control	SUB90	2	Did you use a rule?	Yes
1	Control	SUB90	2	If Yes, I accepted an altitude request when	if the amount of fuel was greater and the plane was smaller I accepted altitude requests

A.4.2 Debrief Analysis

AMBR 3 Debrief Analysis Category 1

Sub-ject	P(E) Trial 8	Correct Accept	Correct Accept	Correct Accept	Correct Accept	Said used rule?	Strategies that Worked Perfectly	Strategies that didn't work perfectly
7	0	1				y	feature	
8	.19	20 L 1	20 S 1	40 L 1	40 S 1	y		1-Feature AND 2 other correlated dimensions
9	0					y	feature	
16	0	20				y	f	
17	0					y	f	
18	0					y	f	
25	0	S				y	f (backwards)	
26	0					y	f	
27	0					y	f	
34	0	S				y	f	
35	0					y	f	
36	0					y	f	
43	0	1				y	f	
44	0					y	f	

45	0					y	f	
52	0	1				y	f	
53	.25					y	f	
54	.13	20 L 1	20 S 1	40 L 1	40 S 1	y		2-feature rule (backwards)
61	.0	L				y	f	
62	0					y	f	
63	0					y	f	
							f	
70	0	S				Y		
71	0					y	f	
72	0					Y	f	
79	.06	3				Y		F
80	.38					N		Incorrect single feature rule
81	0					y	f	
88	0	S				y	f	
89	0					y	f	
90	.06	20 S 1	20 S 3	40 S 1	40 S 3	y		1 2-feature rule
				Correct rule				

Category 3

Subject	P(E) Trial 8	Correct Accept	Correct Accept	Correct Accept	Correct Accept	Said used rule?	Strategies that Worked Perfectly	Strategies that didn't work perfectly
1	.56	20 L 1	20 S 1	40 S 1	40 S 3	yes		"remembered"
2	.06	**20 L 1**	**20 S 1**	**40 S 1**	40 S 3			
		Correct instance	Correct rule		Incorrect rule	yes		1 correct and 1 incorrect 2-feature rule, 1 instance
3	.00	*20 L 1*	*20 S 1*	**40 S 1**	**40 S 3**	yes		
		Correct rule		Correct rule			2 2-feature rules	
10	.25	20 L 1	20 S 1	20 S 3	40 L 1			
	reject	20 L 3	**40 L 3**	40 S 1	**40 S 3**	yes		
		Incorrect rule	Correct rule					"instinct" 1 correct and 1 incorrect 2-feature rule
11	.00	**20 L 1**	**20 S 1**	**20 S 3**	**40 L 1**	yes		
		Correct instance	Correct instance	Correct instance	Correct instance		Memorize 4 instances	
12	.00	20 L 1	20 S 1	20 S 3	40 L 1			
	reject	**20 L 3**	**40 L 3**	*40 S 1*	*40 S 3*	yes		
		Correct rule		Correct rule			2 2-feature rules	
19	.19	20 S 1	40 L 3	40 S 1	40 S 3	yes		1 insufficient correlate
		Insufficient. correlate						
20	0	**20 S 1**	40 L 3	**40 S 1**	**40 S 3**	yes		
		Correct rule with 2 exceptions	Correct Exception				Single- feature rule with 2 exceptions	
	reject	20 L 1	20 L 3	20 S 3	40 L 1			
				Correct Exception				
21	0	*20 S 1*	40 L 3	*40 S 1*	**40 S 3**	yes		
		Correct rule	Correct rule				2 2-feature rules	
28	.31	20 L 1	**20 L 3**	20 S 1	40 L 3	no		

			Correct instance					"memorized" gave 1 instance
29	0	**20 L 1**	20 L 3	**20 S 1**	**40 L 3**	yes		
		Correct rule		Correct instance	Correct instance		1 2-feature rule; 2 instances	
30	0	*20 L 1*	***20 L 3***	20 S 1	40 L 3	no	"guessed"	
		Correct ——— rule		Correlate				1-2 feature rule; 1 Correlate
37	.38	20 L 1	20 L 3	40 L 1	40 S 1	no		"instinct"
38	.31	**20 L 1**	**20 L 3**	40 L 1	**40 S 1**	yes		
.	accept	Correct rule			Correct exception			1 2-feature rule
	reject	**20 S 1**	**20 S 3**	40 L 3	**40 S 3**			and
		Correct rule with 1 exception		Incorrect rule				1 single feature rule with 1 exception, and 1 incorrect 2-feature rule
39	0	**20 L 1**	**20 L 3**	**40 L 1**	**40 S 1**	yes		
		Correct rule (story)	Correct instance (story)		Correct exception (story)		1 2-feature rule, 1 instance,	
	reject	**20 S 1**	**20 S 3**	40 L 3	**40 S 3**		and	
		Correct rule with 1 except (story)					1 single-feature rule with 1 exception	
46	.38	20 L 1	40 L 1	40 L 3	40 S 3	no		"almost subconsciously"
47	.31	20 L 1	40 L 1	40 L 3	40 S 3	yes		1 incorrect instance
	reject	20 L 3	20 S 1	20 S 3	40 S 1			
				Incorrect instance				
48	.13	**20 L 1**	**40 L 1**	40 L 3	40 S 3	yes		
		Correct instance	Correct instance	Correct rule	Correct rule			1 2-feature rule and 2 instances
55	.00	20 L 1	40 L 1	40 L 3	40 S 3	yes		
Same Stimuli as previous group		Correct exception	Correct rule with 2 exceptions (but only 1 mentioned)				1 feature rule with 2 exceptions (but only 1 mentioned); "remembered"	"
56	.06	20 L 1	40 L 1	40 L 3	40 S 3	yes		"remembered"
57	.00	*20 L 1*	*40 L 1*	*40 L 3*	**40 S 3**	yes		

		Rule with 1 exception			Correct instance (story)		1 single-feature rule with 1 exception; 1 instance (story)	
	reject	20 L 3	**20 S 1**	**20 S 3**	40 S 1		and	
		Correct exception	Correct rule				1 2-feature rule	
64	.25	**20 L 3**	20 S 1	**20 S 3**	40 L 3	yes		1 2-feature rule (story)
		Correct rule (story)						
65	.00	**20 L 3**	**20 S 1**	**20 S 3**	**40 L 3**	yes		
		Correct rule	Correct instance		Correct instance		1 2-feature rule and 2 instances	
66	.00	20 L 3	**20 S 1**	20 S 3	**40 L 3**	yes		
		Correct instance	Correct instance	Correct instance	Correct instance		4 memorized instances	
73	.00	*20 L 3*	*20 S 3*	**40 S 1**	**40 S 3**	yes		
		Correct rule		Correct rule			2-2-feature rules	
74	.19	20 L 3	**20 S 3**	**40 S 1**	**40 S 3**	yes		
			Correct instance	Correct rule				1 2-feature rule and 1 instance
75	.38	20 L 3	20 S 3	40 S 1	40 S 3	yes		
		Insufficient correlate						Insufficient correlate
82	.00	**20 L 1**	**20 L 3**	**20 S 1**	**40 L 3**	yes		
		Correct instance	Correct rule	Correct instance			1 2-feature rule and 2 instances	
83	.06	20 L 1	**20 L 3**	20 S 1	40 L 3	no		
			Correct instance					"memorized"
	reject	**20 S 3**	40 L 1	40 S 1	40 S 3			2 instances as examples
		Correct instance						
84	.00	20 L 1	20 L 3	20 S 1	40 L 3	yes		
	reject	***20 S 3***	**40 L 1**	**40 S 1**	*40 S 3*			
		Correct rule	Correct rule				2 2-feature rules	

Category 6

Subject	**P(E) Trial 8**	**Correct Accept**	**Correct Accept**	**Correct Accept**	**Correct Accept**	**Said used rule?**	**Strategies that Worked Perfectly**	**Strategies that didn't work perfectly**
4	.38	20 L 1	20 S 3	40 L 3	40 S 1	yes		Incorrect 2-feature rule

			Incorrect rule					
5	.56	20 L 1	20 S 3	40 L 3	40 S 1	no		Tried incorrect 2-feature rule, then picked "randomly"
6	.25	20 L 1	20 S 3	40 L 3	40 S 1	yes		
	reject	20 L 3	**20 S 1**	**40 L 1**	40 S 3			
			Incorrect Instance2 (reported as "accept" (seen as "opposite" 40 L 1)	Correct instance1				1 correct and 1 incorrect instance, seen as "opposites" (lo lo vs. hi hi?) **Opposites strategy**
13	.56	20 L 1	20 S 3	40 L 3	40 S 1	yes		Smiley faces
14	.00	20 L 1	20 S 3	40 L 3	40 S 1	no	remembered	
15	.00	**20 L 1**	**20 S 3**	**40 L 3**	**40 S 1**	yes		
		Correct pattern rule					Pattern rule 40: hi hi or lo lo 20: lo hi or hi lo	
22	.44	20 L 1	20 S 3	40 L 3	40 S 1	no		"memorize"
23	.00	**20 L 1**	**20 S 3**	**40 L 3**	**40 S 1**	yes		
		Correct instance-4	Correct instance-3	Correct instance1	Correct instance2		Memorize 4 instances; "got the pattern" (unspecified)	
24	.56	20 L 1 Correct exception	**20 S 3**	**40 L 3**	40 S 1 incorrect 2-feature rule	yes		Incorrect 2-feature rule
	reject	20 L 3	20 S 1	40 L 1	40 S 3			and
				Correct 2-feature rule with 1 exception				Correct 2-feature rule- with – exception
31	.38	20 L 1	20 S 3	40 L 3	40 S 1	no		"guessed"
32	.38	20 L 1	20 S 3	40 L 3	40 S 1	no		
					Incorrect 2-feature rule			Incorrect 2-feature rule
	reject	**20 L 3**	**20 S 1**	40 L 1	40 S 3			and
		Correct instance (story)	Correct instance (counter-					Two correct instances

			intuitive)					
33	.00	**20 L 1**	**20 S 3**	**40 L 3**	**40 S 1**	yes		
		Correct pattern rule					Pattern rule L: hi hi or lo lo S: lo hi or hi lo	
40	.50	20 L 1	20 S 3	40 L 3	40 S 1	no		remember
41	.00	**20 L 1**	**20 S 3**	**40 L 3**	**40 S 1**	yes		
		Correct pattern rule					Pattern rule 3: hi hi or lo lo 1: lo hi or hi lo	
42	.19	20 L 1	20 S 3	40 L 3	40 S 1	no		remember
49	.00	**20 L 1**	**20 S 3**	**40 L 3**	**40 S 1**	yes		
		Correct instance	Correct instance	Correct instance	Correct instance		4 memorized instances	
	reject	***20 L 3***	20 S 1	40 L 1	***40 S 3***		Started with	
			Correct instance	Correct instance			2 memorized instances	
50	.50	20 L 1	20 S 3	40 L 3	40 S 1	yes		1 instance
			Correct instance					
51	.50	20 L 1	20 S 3	40 L 3	40 S 1	no		
					Incorrect 2-feature rule (story)			Irrelevant feature (locale); Incorrect 2-feature rule (accept story)
	reject	**20 L 3**	**20 S 1**	**40 L 1**	**40 S 3**			And
		Incorrect 2-feature rule (story)						Incorrect 2-feature rule (reject story)
58	.44	**20 L 3**	20 S 1	**40 L 1**	40 S 3	yes		3 correct instances and lots of incorrect rules tried
		Correct instance						
	reject	**20 L 1**	**20 S 3**	40 L 3	40 S 1			
		Correct instance	Correct instance (story)					
59	.00	**20 L 3**	**20 S 1**	**40 L 1**	**40 S 3**	yes		
		Correct pattern rule					Pattern rule: 20: hi hi or lo lo	

							40: hi lo or lo hi	
60	.56	20 L 3	20 S 1	40 L 1	40 S 3	yes		Irrelevant feature ("clear of other planes")
67	.5	20 L 1	20 S 3	**40 L 3**	40 S 1	no		1 instance
				Correct instance (story)				
68	.25	20 L 1	20 S 3	**40 L 3**	40 S 1	yes		
		Incorrect 2-feature rule						Incorrect 2-feature rule
69	.38	20 L 1	20 S 3	**40 L 3**	40 S 1	yes		Unexplained rule (2 features)
	reject	20 L 3	20 S 1	40 L 1	40 S 3			and
				Incorrect instance "accept"				Incorrect instance
76	.5	20 L 3	20 S 1	40 L 1	40 S 3	no		
				Correct instance	Incorrect 1-feature rule			Incorrect 1-feature rule; 1 memorized instance
77	.5	20 L 3	20 S 1	**40 L 1**	**40 S 3**	yes		
				Incorrect 1-feature rule				Incorrect 1-feature rule
78	.63	20 L 3	20 S 1	40 L 1	40 S 3	no		memorization
85	.19	**20 L 1**	20 S 3	40 L 3	40 S 1	no		
		Correct instance						Memory; 1 correct instance (accept)
	reject	20 L 3	**20 S 1**	40 L 1	40 S 3			And
			Correct instance					1 correct instance (reject)
86	.00	**20 L 1**	**20 S 3**	40 L 3	**40 S 1**	yes		
		Correct instance 3	Correct instance 1		Correct instance4		Referring to pattern rule? Order of instances 1-4 suggests: 3: Lo lo accept Hi lo reject 1: Lo hi accept Hi lo accept **Opposites Strategy**	

	reject	20 L 3	**20 S 1**	40 L 1	**40 S 3**			
					Correct instance2			
87	.00	**20 L 1**	**20 S 3**	**40 L 3**	**40 S 1**	yes	Memorized 4 instances	

Summary of Results

Perfect Scorers (Trial 8)				
(Simple Strategies)	**Category 1**	**Category 3**	**Category**	
1-Feature Rule 1*	(24 Ss)			Rule
1-Feature Rule With 2 Exceptions 3*		#20		Rule + Exception
1-Feature Rule with 1 Exception (forgot to mention 2nd exception?)		#55		Rule + Exception
2 2-Feature Rules 3*		#3, #12, #21, #73, #84		Rule
2-Feature Rule; 2 Memorized Instances 3*		#29, #48, #65, #82,		Rule + Instance
2-Feature Rule; 3* 1 Correlate not always		#30		
Pattern Rule 6*			#15, #33, #41, #59	Rule
(Complex Strategies)				
Accept 1 2-Feature Rule; 1 Memorized Instance Reject: 1-Feature Rule with 1 Exception 3*		#39		Rule + Exception + Instance
Accept: Single-Feature Rule with 1 Exception; 1 Memorized Instance Reject: 2-Feature Rule 3*		#57		Rule + Exception + Instance
Memorize 4 instances 1*3*6*		#11, #66	#23, #49, #86 #87	Instance
(Yes to Rules Question)		(14 Ss)	(6Ss)	
(No to Rules Question)		#30	#14	
Imperfect Scorers Trial 8				
(Simple Strategies)				
Single feature rule	#79, #53		#77	Rule
Single feature rule-incorrect	#80			Rule
Single feature rule-incorrect And 1 memorized instance			#76	
Single Feature Rule (irrelevant)			#60	Irrelevant Rule
Single Feature AND 2 other (insufficient) correlated dimensions Rule	#8			Rule
Single Feature Rule (irrelevant);			#51	Rule

2 2-Feature Rules-incorrect				
2-Feature Rule		#64		Rule
2-Feature Rule -incorrect	#54, #90		#4, #5,#68	Rule
2-Feature Rule; 2 Memorized Instances 3*		#48		Rule + Instance
2-Feature Rule (incorrect); 2 memorized Instances			#32	Rule + Instance
2-Feature Rule; 1 Memorized Instance		#74		Rule + Instance
2-Feature Rule (incorrect); 1 memorized Instance (incorrect)			#69	Rule + Instance
2 2-Feature Rules (one was incorrect); 1 Memorized instance		#2		Rule + Instance
2 –2 Feature Rules (one was incorrect)		#10		Rule
1 Memorized Instance		#28	#50, #67	Instance
2 Memorized Instances			#85	Instance
1 Memorized Instance (incorrect)		#47		Instance
2 Memorized Instances (one was incorrect)			#6	Instance
1 Insufficient Correlate		#19, #75		Rule
Rule: Single feature (correct) AND 1 correlate (insufficient)				
"Yes" to rule with content		#1,#2,#10,#19,#38, #47,#64,#74,#75	#4,#6,#24,#50,#58, #60.#68,#69,#77	
"Yes" to rule, but no content (e.g., "remembered, "guessed," "almost subconsciously")		#1, #46, #56	#13	
"No" to rule with no content (guessed, instinct)		#37, #46, #83	#22,#31,#40,#42	
"No" to rule, but had content		#28,	#5, #32,#51, #76	
(Complex Strategies)				
Accept: 2-Feature Rule Reject: 1-Feature rule with 1 Exception; 2-Feature Rule (incorrect)		#38		Rule + Exception
Accept: 2-Feature Rule (incorrect) Reject: 2-Feature Rule with 1 Exception			#24	
No single strategy– Multiple Strategies described			#58	

SIMPLIFIED CHART

Perfect Scorers (Trial 8)				
	Category 1	**Category 3**	**Category 6**	
Rule	(24 Ss)	#3, #12, #21, #73, #84	#15, #33, #41, #59	
Rule + Exception		#20,#55		
Rule + Instance		#29, #65, #82,		
Rule + Correlate		#30		
Rule + Exception + Instance		#39,#57		
Instance (s)		#11, #66	#23, #49, #86, #87	
Memorized (but no instances given)			#14	
(Yes to Rules Question)	(24 Ss)	(14 Ss)	(8)	
(No to Rules Question)		#30	#14	
Imperfect Scorers Trial 8				
Rule	#8, #53, #54, #79, #80,#90	,#10, #19, #64, #75	#51, #68, #24	
Rule + Instance		2, #48,#74,	#32, #58, #69, #76	
Rule + Exception		#38		
Irrelevant Rule			#60,#77	
Instance (s)		#28,#47	#6,#50, #67, #85	
Memorized (but no instances given)		#1, #37, #46, #56, #83	#13, #22,#31,#40,#42, #78	
"Yes" to rule with content	#8, #53, #54, #79, #90	#1,#2, #10, #19, #38, #47, #48,,#64, #74,#75	#4,#6,#24,#50,#58, #60.#68,#69,#77	
"Yes" to rule, but no content (e.g., "remembered, "guessed," "almost subconsciously")		#1, #56	#13	
"No" to rule with no content (guessed,		#37, #46, #83	#22,#31,#40,#42, #78	

instinct)				
"No" to rule, but had content	#80	#28,	#5, #32,#51	

SIMPLIFIED CHART

Perfect Scorers (Trial 8)				
	Category 1	**Category 3**	**Category 6**	
Rule	(24 Ss)	#3, #12, #21, #73, #84	#15, #33, #41, #59	
Rule + Exception		#20,#55		
Rule + Instance		#29, #65, #82,		
Rule + Correlate		#30		
Rule + Exception + Instance		#39,#57		
Instance (s)		#11, #66	#23, #49, #86, #87	
Memorized (but no instances given)			#14	
(Yes to Rules Question)	(24 Ss)	(14 Ss)	(8)	
(No to Rules Question)		#30	#14	
Imperfect Scorers Trial 8				
Rule	#8, #53, #54, #79, #80,#90	,#10, #19, #64, #75	#51, #68, #24	

Rule + Instance		2, #48,#74,	#32, #58, #69, #76	
Rule + Exception		#38		
Irrelevant Rule			#60,#77	
Instance (s)		#28,#47	#6,#50, #67, #85	
Memorized (but no instances given)		#1, #37, #46, #56, #83	#13, #22,#31,#40,#42, #78	
"Yes" to rule with content	#8, #53, #54, #79, #90	#1,#2, #10, #19, #38, #47, #48,,#64, #74,#75	#4,#6,#24,#50,#58, #60.#68,#69,#77	
"Yes" to rule, but no content (e.g., "remembered, "guessed," "almost subconsciously")		#1, #56	#13	
"No" to rule with no content (guessed, instinct)		#37, #46, #83	#22,#31,#40,#42, #78	
"No" to rule, but had content	#80	#28,	#5, #32,#51	

Appendix B: AMBR Paper (Presented at BRIMS, 2003)

The AMBR Project: A Case-Study in Human Performance Model Comparison

Yvette J. Tenney
David E. Diller
Richard W. Pew
Katherine Godfrey
Stephen Deutsch
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
ytenney@bbn.com, ddiller@bbn.com,
pew@bbn.com, kgodfrey@bbn.com, sdeutsch@bbn.com,

Keywords:
human performance modeling, human behavior representation, cognitive architecture, model validation, air traffic control, concept learning

ABSTRACT: *The Agent-Based Modeling and Behavior Representation (AMBR) Program, sponsored by the Air Force Research Laboratory, was designed to advance the state of the art in cognitive and behavioral modeling in domains of relevance to the military. The project has funded four rounds of model development concerned with air traffic control tasks. In each round, multiple developers created different models of the same human operator activities and were compared to human participants performing the same tasks in a non-competitive "fly off." The tasks required memory, learning, multitasking, and interruption handling, as well as basic perceptual and motor processes. BBN Technologies acted as moderator for the model comparison. CHI Systems, Soar Technology, Carnegie Mellon University, and the Air Force Research Laboratory developed the models. In this paper, we describe the general approach and methodology of AMBR. We then summarize the lessons learned. Collectively, we found better ways of illuminating the essential elements of the models. We evolved more rigorous tests of the models—Transfer tests were used to drive the modelers to predict behaviors. Finally, the multiple development and workshop phases fostered the migration of important modeling techniques across teams. We conclude by extrapolating from the AMBR project to the model procurement process with suggestions on how to promote the development of better human performance models.*

Introduction

The sustained interest among DoD, NASA and other agencies in more robust, realistic human performance models (HPMs) for use in simulations for training and system acquisition leads us to seek R&D strategies that will result in higher quality models [1]. Our experience suggests that there are no short cuts to better models. There is always a need for (1) more detailed data about the behaviors being modeled, (2) greater understanding of the fundamentals of human performance that can be incorporated into models, (3) improved architectures in which to build models and (4) better methods for verification and validation.

The Agent-Based Modeling and Behavior Representation (AMBR) Project, sponsored by

the Air Force Research Laboratory, has provided an opportunity for multiple developers to create different models of the same human operator activities and to compare the results both from model to model and with human participants performing the same tasks. AMBR provided a forum in which to identify the needs identified above and to collectively make substantive progress on each of them.

The project has funded four rounds of model development and validation. BBN Technologies has had the role of model comparison moderator. CHI Systems, Soar Technology, Carnegie Mellon University and the Air Force Research Laboratory have been the model developers in these highly collaborative studies. As the project nears completion, we at BBN wish to share with the HBR community the distinctive features of the program, the lessons learned and what we believe are important implications for future procurement of human performance models.

In the sections that follow we will describe the general approach and methodology utilized in each phase of the project, present some typical results and then summarize the lessons learned with respect to responding to the needs identified in the first paragraph. Finally, we will abstract from this framework the recommendations that we wish to make to future procurers of human performance models (HPMs).

AMBR Goals

The AMBR Project was designed to advance the state of the art in cognitive and behavioral modeling. Specifically, the program encouraged the development of models of complex tasks in domains relevant to military applications. It was important that the domain required models of integrative performance, requiring the coordination of memory, learning, multitasking, interruption handling, and the perceptual and motor systems in order to scale more effectively to real-world environments. Furthermore, the program provided a structure by which models could be developed and judged by their ability to be predictive, rather than only descriptive or explanatory. Additionally, the program collected new data of interest to the human performance modeling community at large and is making these data and the testbed available to the modeling community by creating a repository of the simulation environments and human performance data.

AMBR Framework

Roles and Participants

The AMBR Program was organized as a series of comparisons among alternative modeling approaches with each comparison focused on a set of cognitive/behavioral capabilities that taken together allow for the creation of more complete and integrative models within the chosen task domain. Each comparison involved four human performance models: the Air Force Research Laboratory's DCOG [2], Carnegie Mellon University's ACT-R [3], CHI Systems, Inc.'s COGNET/iGEN [4], and Soar Technology's EPIC-Soar [5].

The modeling teams were extremely diverse in their modeling approaches, architectural frameworks, and underlying theoretical assumptions upon which the models were built. Two of the modeling teams emphasized the development of models within a constrained set of previously established core theoretical constructs, while other teams were much more flexible in their ability to represent cognitive processes at higher levels of description. Three of the modeling teams were building upon established systems with documented track records, while one modeling team built a new model, DCOG, designed exclusively as part of the AMBR program. Modeling teams consisted of university research projects, government research projects, and commercial enterprises.

Model comparisons were organized and overseen by a neutral moderator, BBN Technologies. The moderator, in cooperation with the government, determined the task domain and behaviors to be modeled in each phase of the program. Critical requirements for the chosen behaviors were: a) there are no adequate models that already perform the selected behaviors, and b) creating such a model in the chosen domain challenges prevailing modeling approaches. Once the task domain and behaviors were chosen, the moderator designed and developed a simulation of a task involving the specified behaviors. To make it experimentally feasible to elicit and collect the behaviors of interest, the task was simplified in ways that preserved the essential elements. A main requirement for this simulation was that it could be "operated" either by real humans or by HPMs. After the simulation was developed, the moderator collected data from humans performing the task.

A panel of experts, solicited from the HPM community, was invited to review and evaluate the results of each round of model comparisons. The panel was chosen to include members with expertise in one or more of the following: the modeling architectures used by the modeling teams; the cognitive and behavioral phenomena to be addressed by that round of model comparisons; other modeling efforts within the government and the military.

Phased Project Structure

The AMBR Program has supported four rounds or phases of model development and validation. Although each phase focused on different behavioral phenomena, the phases were designed to build upon previous phases of model development. In this way, models were developed incrementally, with modeling teams sharing insights at each phase of development.

The structure and process of the model comparison evolved significantly over the course of several rounds of model comparisons. The process that we found to be successful for developing and comparing models can be outlined as follows:

1. The moderator team, in cooperation with the government sponsor, identifies the modeling goals for this phase of the program and determines what cognitive/behavioral phenomena are to be stressed.
2. The moderator and sponsor select a task domain that emphasizes the capabilities identified in step 1 and is relevant to military modeling and simulation needs. It is useful at this phase of the process to consider the types of modeling architectures involved in the comparison processes, their current capabilities, and how the task domain should challenge and stretch current modeling capabilities.
3. The moderator in conjunction with the sponsor and the modeling teams adapts or designs a task within the chosen domain which facilitates the elicitation of the desired behavioral phenomena as part of a formal experiment. The types of data useful to the modeling teams must be determined, as well as the measures to be used in the comparison between models.
4. The moderator borrows, modifies or constructs a simulation of the task in which both a human-in-the-loop or a human performance model can operate. The simulation is made available to the modeling teams.
5. The moderator collects, analyzes, and disseminates human performance data by running an experiment with human participants. The moderator may choose to withhold a portion of the human performance data in order to provide the modeling teams with an opportunity to predict human performance.
6. The modeling teams develop models that attempt to replicate human performance when performing the task and predict human performance on portions of the task for

which human performance data were not released to the modelers.

7. The moderator team provides data comparing the performance of the models to the human data.

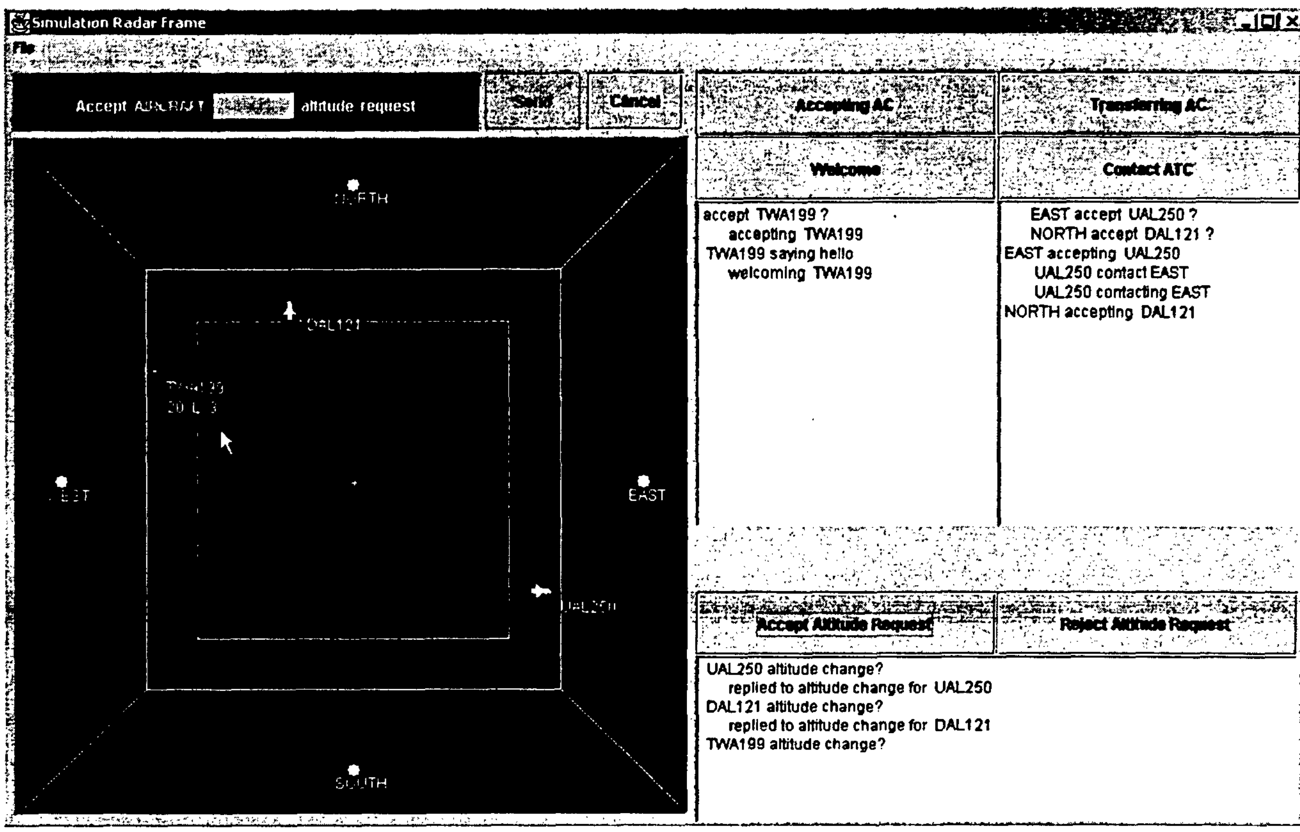


Figure 1: Air Traffic Display Task

8. The expert panel convenes to compare and contrast the developed models and the underlying architectures supporting them. Data previously withheld from the modeling teams is provided and compared to the predictions made by the models.
9. Steps 6, 7, and 8 are repeated for any data previously withheld from the modeling teams.

Simulation Environment and Task Domain

The AMBR task domain is a simplified version of en-route air traffic control [6]. A primary air traffic control sector is displayed together with the boundaries of four adjoining sectors (See Figure 1). The participant in the experiment, either a human or a model, plays the part of an air traffic controller in charge of a sector of airspace. During an experimental trial the central controller must manage aircraft that are arriving and departing from the adjoining sectors. This involves communication with the aircraft and with the adjoining sector controllers. This communication is accomplished by radio button presses on the display rather than by voice. When a simulation trial is run, a complete, time-stamped trace of the time course of every action by the controller (either human or model) is recorded in a history file. These data may be used to derive any desired individual or aggregate measure of performance.

AMBR Phases 1 & 2: Multi-tasking

The emphasis in Phases 1 and 2 was on the basic air traffic control situation. This situation was of interest because of the potential for human operator overload and the need for

effective information management strategies. The goal was to foster understanding of multi-tasking strategies, a capability not widely available in existing models, while providing a relatively straightforward task for "shaking down" the models. In Phase 1, three workload levels (2, 3, or 4 aircraft per minute) were tested with two display conditions— a Text condition in which all messages had to be read and a Color condition in which color codes signaled the action required and obviated the need for reading. Each participant experienced each of the nine display and workload conditions in scenarios lasting approximately 10 minutes. Penalty points were awarded for incorrect, delayed, or missed actions. As seen in Figure 2, with respect to penalty points, all of the models showed effects of display and workload conditions. However, none of the models met the strict criterion of falling within the confidence intervals for the human data in all conditions.

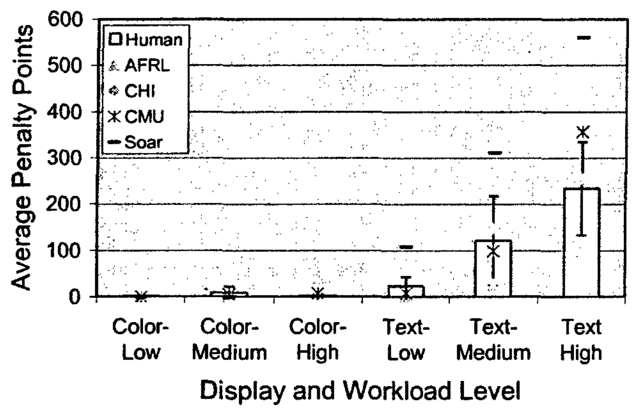


Figure 2: Penalties as a Function of Display and Workload

Phase 2 replicated and extended the Phase 1 results by providing a comparison of human and model performance in both HLA and Non-HLA environments [2-6, 7]. The type of environment made no difference in the performance of either the humans or the models.

AMBR Phases 3 & 4: Category Learning Phases 3 and 4 of AMBR involved a classic concept-learning task embedded within the basic air traffic control situation. Subjects had to learn to make correct decisions to accept or reject altitude change requests, based on three bi-variate properties of the aircraft (percent fuel remaining, aircraft size, and turbulence level). A novel feature of the experiment was the addition of multi-tasking to this concept learning paradigm. In addition to the altitude change requests (the concept learning task), the participant had to hand-off a number of aircraft to adjoining controllers (secondary task).

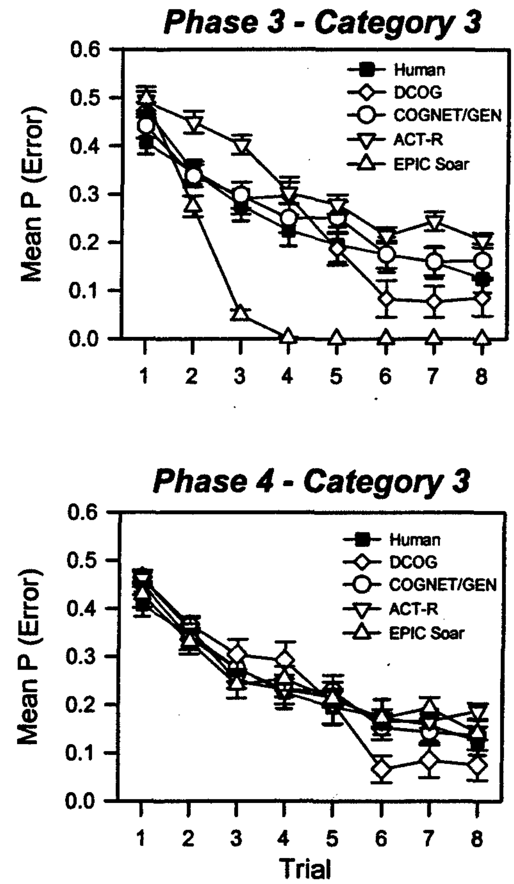


Figure 3: Participant and Model Performance in Phases 3 and 4 in the Category Type 3 Learning Task

The design consisted of 9 conditions, defined by 3 category structures and 3 workload levels. Three of the six Shepard, Hovland, and Jenkins category structures [8] were used: single attribute relevant (Type 1), a single-attribute rule plus exceptions (Type 3), and no rule (Type 6). These category structures were of interest because the pattern of results has historically been problematic for both rule-based and

instance-based systems. While we would have preferred a learning task that was more realistic in the context of air traffic control, the large literature and rich set of findings associated with this task appealed to us. The three workload levels consisted of 0, 12, or 16 required handoffs, in addition to the 16 altitude requests. There were 8 scenarios, or trials, lasting ten minutes each. After the 8 learning scenarios, participants performed a 'transfer' scenario; choosing to accept or reject altitude change requests for aircraft with novel property values. One hour of training on the mechanics of the tasks preceded the trials.

Each modeling team ran their human performance models one or more times in each condition (Phase 3) and then had a chance to revise them if they wished (Phase 4). As expected, humans learned Category Type 1 faster than Category Type 3 and learned Category Type 3 more quickly than Category Type 6. All of the models also showed this general trend. There was, however, a large degree of variability in how well different models quantitatively matched the data (See Figure 3, Phase 3). It is important to note that the matches improved when the modelers were given the opportunity to tune and revise their models based on ideas that emerged in the comparison workshop (See Figure 3, Phase 4).

Surprisingly, the ability to predict the results of the transfer scenario was a significant challenge to the modelers. None of the modeling teams fit the transfer data either to their or our satisfaction. In particular, none of the models predicted the significant drop in performance from the last training scenario to the transfer trials for the items that were identical in both sets of trials ('Trained' vs. 'Original Trial 8'). Additionally, no model predicted the decrease in performance found for 'extrapolated' items relative to previously seen items (See Figure 4, Phase 3 which shows data averaged across the four models).

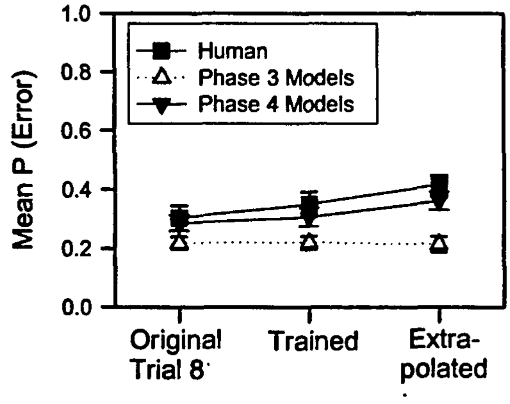


Figure 4: "Typical" Phase 3 & 4 Transfer Task Results for Category Type 6

Modelers were then provided with the transfer scenario data and a final round of model comparisons was performed. Modelers revised their models, significantly improving their fit, both to the transfer data and the concept learning data. (See Figure 4, Phase 4).

Model Adaptations Across Project Phases

By adding a category learning component to the air traffic control task, modelers were required to either activate and adapt learning algorithms already existing in their models, or in several cases, develop an entirely new learning mechanism. The degree to which learning mechanisms were integrated into existing model architectures varied widely, ranging from a separate sub-module implemented as a 'black box' and able to be manipulated independently from the rest of the system, to a fundamental component and constraint on the model architecture itself.

Modelers were provided with two opportunities to revise their models during phases 3 & 4, with most modeling teams making a significant number of changes to their models. Changes were made at the levels of adjusting parameter values, revising or replacing the basic learning algorithm, and modifying the model architecture.

Most teams carried over parameter values used in previous phases of the project, and added and

adjusted parameters as necessary. For most teams, this meant revising a significant number of their parameters values, although one team revised only a single parameter value. Several teams added additional strategies and mechanisms for learning in order to account for the range in performance found across category types as well as between individuals. Additionally, refinements were made to various model components such as working memory and visual feature extraction in order to account better for human performance.

Model Comparison Workshops

A workshop was held at the conclusion of each round of model comparison. Each workshop provided a venue for the program participants to discuss the results of that round of model comparison and to plan the next round of comparisons. Modelers were asked to describe their models by 1) providing an overview of their models and a description of how they address the target problem, 2) describing the theory and architecture on which the model is based, as well as the assumptions and intuitions underlying it, 3) describing the unique features of their model, and 4) discussing the unique challenges of this task and how they were handled, as well as the successes and failures of the model for this task.

After presentations by the modelers, the expert panel provided a summary of the strengths and weaknesses of each model, commenting on how well each model did at replicating and predicting the data. Additionally, the panel discussed issues, challenges, and recommendations for future rounds of the project.

Lessons Learned

Throughout the AMBR project, the future of human performance modeling was a concern very much in the forefront of every participant's mind. Our first concern was to find new and productive means by which to move the state of the art forward. Making the lessons learned broadly available is an important step in that direction.

How to Communicate about Models

One of the first lessons we learned was the difficulty of communicating about model characteristics among team members. There is no question that it is difficult to describe the intricacies of a human performance model to people in different disciplines. In AMBR Phase 1, the Expert Panel felt that they did not have enough information or background to undertake the kind of evaluation intended, although they had listened to each of the modelers talk for two hours [9]. In AMBR Phase 3, a member of the audience at the AMBR Cognitive Science Symposium, remarked on how difficult it was to grasp the subtle distinctions between models, when most people are familiar with only one. Yet, as the AMBR Program has progressed, communication has improved considerably.

Several things, other than the benefits of continued exposure, helped bridge the communication gap, making it easier for the Modelers, the Moderators and the Expert Panel to understand and appreciate each other. The aids to communication we settled upon constitute one of our most important lessons learned. The first is that it is easier to get a feeling for a model when the interface contains some indication of what the model is doing while it is "running." Although this aspect of model development was not emphasized in AMBR, the interface for the SOAR model in AMBR Phase 1 provided insight into the model's eye movements and mental state, in a unique and dramatic way. Secondly, it helps to hear a "walk through" of the task. In AMBR Phase 3 each of the modelers described how the model learned to respond to the concept task, step-by-step. Third, it is important to hear how each parameter value was selected (e.g., from the development data, from the human performance literature, from previous modeling efforts, or by trial and error). In AMBR Phase 3, the Panel took pains to elicit this information

when it was not offered spontaneously. Fourth, it is interesting to see how different modelers choose to focus on different aspects of the common set of human data, to support their modeling approach. Finally, it is particularly useful to hear how specific changes to the model either did or did not improve the fit. In AMBR Phase 4, several of the modelers talked about a series of mini-experiments, in which they made changes and iteratively ran their model. These descriptions brought the models to life and conveyed the feeling that humans may someday have as much to learn from models as models are learning from humans.

Importance of Collecting a Rich Set of Human Data

The second lesson is the realization of how critical the human data collection effort has been to the project. At the same time, we have had to learn that it is possible to surmount setbacks that can befall the best intentioned data collection efforts. In AMBR Phase 3, we had hoped to expand the literature on concept learning into the realm of dual-task performance, but were thwarted in this effort when our workload manipulation, though effective in increasing errors in the secondary task, failed to have an effect on the category learning task. What we found, fortunately, is that even without an effect of secondary task workload, our concept learning data were rich enough to challenge the modelers. There was enough for the modelers to work with for a number of reasons, which constitute important lessons learned. First, we collected data on a number of different kinds of measures—accuracy, reaction time, subjective workload, and subjective strategy reports. Second, we tested enough subjects to yield a range of individual differences. Third, we collected data on different phases of the task, e.g., a concept learning phase and a transfer test (involving identical, near-transfer, and far-transfer items). In fact, in retrospect, it would have been even better if we had included a pretest (e.g., measuring personality and skill level) that could have served as the basis of classifying subjects into "bins" on an a priori basis. Such data could have aided the DCOG modelers, who use personality variables in their model (e.g., patience and tolerance) as a determinant of individual differences in concept learning. The moral is that, within budget and time constraints, the more "beefed" up the design, the better, from the point of view of having the critical data needed for sophisticated modeling. The downside is that designing, running, and analyzing a complex study takes considerable time.

It is important to put sufficient resources into the human data collection effort. Pilot testing will optimize the chances of obtaining meaningful data. To get the best advice about the experimental task and dependent measures that will be helpful for individual models, include the modelers and expert panel in the planning phase. If possible, jump-start the modeling effort, while the human data are being collected, by providing modelers with references to relevant data from the literature. We referred the modelers to classical concept learning results, which were a good approximation, though not identical, to the human data we collected. Provide the task environment to the modelers and Expert Panel early on, so they can gain insights into strategies by testing themselves or observing others. Provide the raw data to the modelers as well as easily "digested" data summaries, so they can discover aspects of the data that may be uniquely relevant to their own modeling interests. For example, one of the AMBR Phase 3 modeling teams divided subjects into learners and non-learners (based on the data summaries provided) and then counted the number of times each type of subject changed their original answer on identical problems on the transfer test. Another team decided to examine how closely individual model runs matched individual subject runs.

Benefits of a "Fly Off"

After three years of collecting, analyzing, and distributing data, organizing team meetings, and presenting conference papers, we are convinced that the combination of Modelers, Expert Panel and Moderators, working on a "Fly Off," is the right way to inspire innovation. (Our "Fly Off" was non-competitive and should be thought of simply as a model comparison). Starting with AMBR Phase 1 and continuing today, we have seen a remarkable exchange of ideas among the modelers and a willingness to incorporate the best of each. The successive iterations of each "Fly-Off" have also seen a greater accommodation of the models to the human data findings, which will make them both more predictive and explanatory. Finally, we have seen a nice exchange between the modelers and the moderators, with a careful weighing on each side of how much of the human data vs. the model's behavior to trust. Both modelers and moderators have daunting tasks. Collaboration, as demanded by the AMBR Fly-Off framework, can only serve to benefit both.

Extrapolating to Model Procurement

Model procurement will have a singular influence on how we make progress on the difficult goal of making substantive improvements in model performance. We would like to suggest that by extrapolating from the lessons learned in AMBR, we can make a few suggestions with the goal of assuring better outcomes from model procurement. To help focus this brief discussion we will consider the procurement of models to operate in a training environment.

A good starting point is that model procurement, like model development, should be a multidisciplinary activity. It should involve model developers, subject matter experts, operational trainers, and behavioral scientists. Many critical decisions are made during the *preparations* for procurement. Decisions made with respect to operational domain, training objectives, and range of behaviors to be supported by the human performance models each will have a significant impact on the likelihood of a successful procurement outcome. In a multi-step procurement process that allows preliminary discussion and includes white papers before proposals are submitted, there is the opportunity for iterative refinement as these decisions are made—the procurement itself will be refined. Importantly, each competing team can have the best opportunity to favorably impact project direction.

The operational group must put forward its training goals with the expectation that those goals may well be refined, cut back in some cases and improved in others as the procurement process goes forward. Each team should have the opportunity to provide a detailed view of how those training objectives would be addressed: what the scenarios are that will be required; what models are needed and how will they be developed; what is the range of situations that the proposed models will need to be capable of addressing in order to meet the operational training goals. The range of issues to be addressed drives the requirement for multidisciplinary participation on both sides of the procurement. The iterative requirements process initiated during procurement will be essential to successfully adapt to evolving requirements as model development proceeds.

Achieving a successful model procurement process requires good insight into proposed model design, functionality, and behavioral outputs. Our experience in AMBR suggests that this is not an easy process. In the first phase of AMBR, the attempt to get a top-down view into model design and capabilities was largely unsuccessful. Modeling teams were simply not able to make clear the important aspects of their models in a way that enabled the other teams, the moderators, and the expert panel to feel that they had a good grasp of what was important to model execution. In later phases, a "middle-out" approach was requested of the teams. The teams

focused their discussion on the elements of the models critical to the specific task and later on the changes required as improvements were made in subsequent phases. Visibility into the models was markedly improved. Asking for input on model design and capabilities at this level may secure a better view of proposed capabilities.

From the perspective of procurement, the message of the AMBR *transfer* task is not encouraging. Human performance models have generally been fragile at the boundaries of their intended performance envelopes. With this in mind, goals in selecting the transfer task were modest. That modeling teams had modest success at best was consistent with previous experience. There are two points to be made here: the first is that in a procurement for training systems, it is critical that the range of required behaviors be identified early on—transfer is not going to address emerging requirements and the boundaries of those behaviors will be expensive to grow. The second point is that progress on this very difficult research problem (transfer/robustness) has the potential to make a very significant contribution to our basic understanding of human capabilities and in the process, make a significant improvement in the capabilities of our human performance models.

Lastly, individual differences emerged as a significant factor in model development in AMBR Phase 3. The aggregate data were very neat and tidy and nicely matched broadly reported results in the literature. In examining the human subject data, we found a broad range of behaviors that when combined, produced that tidy aggregate behavior. And indeed, one of the features most often missing in the models that have been procured to date is a reasonable range of responses to a given situation. Attention to individual differences has the potential to contribute to improvements in the range of behaviors that models can provide. Procurements can require individual differences as a means to obtain a range of behaviors. Modest, well-placed research efforts can make a difference here too.

Conclusions

The idea of advancing the state of the art in human performance modeling by challenging multiple research groups to try their systems on a common task is not new to AMBR. Fifteen years ago, the Department of Defense enlisted seven teams to compete in building training systems to teach humans to play a battle simulation game, called Space Fortress [10]. The results provided noteworthy insights into the subtleties of training and transfer.

The AMBR human performance modeling project was conceived in this spirit, but has stressed cooperation and innovation rather than competition. By providing the opportunity for coordinated, interdisciplinary, multi-phase development efforts, striking improvements and advances have been made by all the modeling teams. The lessons learned from AMBR are encouraging and should inspire funders interested in advancing the state of the art in human performance modeling, to continue in this direction.

Acknowledgements

We gratefully acknowledge the sponsorship of this research by the Human Effectiveness Directorate of the Air Force Research Laboratory. We thank AFRL Program Managers Mike Young and Kevin Gluck for their guidance and support. We are extremely grateful to the modeling teams for their contributions to the AMBR project, including Bob Eggleston and Katherine McCreight (AFRL), Wayne Zachary, Jim Stokes and Joan Ryder (Chi Systems, Inc.), Christian Lebiere (CMU), and Ron Chong and Robert Wray (Soar Technology, Inc.) We also thank the members of the Expert Panel (Shelly Baron, Gwen Campbell, Wayne Gray, Harold Hawkins, Bonnie John, Brad Love, and Peter Polson) for

their participation in this project. We thank Kevin Gluck for his comments on early drafts of this paper.

References

[1] R. W. Pew & A. S. Mavor: Modeling Human and Organizational Behavior: Application to Military Simulations, National Academy Press, Washington, D. C. 1998.

[2] R. G. Eggleston, M. J. Young, & K. L. McCreight: "Modeling Human Work through Distributed Cognition" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp. 99-103, May 2001.

[3] C. Lebiere, J. R. Anderson, & D. Bothell: "Multi-tasking and Cognitive Workload in an ACT-R Model of a Simplified Air Traffic Control Task" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp. 91-98, May 2001.

[4] W. Zachary, T. Santarelli, J. Ryder, J. Stokes, & D. Scolaro: "Developing a Multi-tasking Cognitive Agent Using the COGNET/iGEN Integrative Architecture" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp. 79-90, May 2001.

[5] R. Chong: "Low-level-Behavioral Modeling and the HLA: An EPIC-Soar Model of an Enroute Air-Traffic Control Task" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp. 27-36, May 2001.

[6] S. Deutsch & B. Benyo: "The D-OMAR Simulation Environment for the AMBR Experiments" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp 7-13, May 2001.

[7] Y. J. Tenney & S. L. Spector: "Comparison of HBR Models with Human-in-the-loop Performance in a Simplified Air Traffic Control Simulation with and without HLA Protocols: Task Simulation, Human Data and Results" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp. 15-26, May 2001..

[8] R. N. Shepard, C. L. Hovland, & H. M. Jenkins: "Learning and Memorization of Classifications" Psychological Monographs, Vol. 75 (13, Whole No.517), 1961.

[9] K. A. Gluck & R. W. Pew: "Overview of the Agent-based Modeling and Behavior Representation (AMBR) Model Comparison Project" Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation, pp. 3-6, May 2001.

[10] M. Fabiani, J. Buckely, G Gratton, M.G.H. Coles, E. Donchin, & R. Logie: "The Training of Complex Task Performance" Acta Psychologica, Vol. 71, pp. 259-299, 1989.

Author Bibliographies

YVETTE J. TENNEY is a cognitive psychologist with interests in the areas of applied cognition, training and human factors. She received her B.A. and Ph.D. in psychology from Cornell University. As a Senior Scientist at BBN Technologies, in Cambridge, MA, she has been involved in developing training technologies and job aids for underwater exploration, harbor navigation, complex electronic troubleshooting, command and control, and commercial aviation. Her recent work has focused on problems of operators in semi-automated, multi-task environments, particularly on the maintenance of situation awareness and the management of cognitive workload.

DAVID E. DILLER is a Senior Scientist at BBN Technologies in Cambridge, MA. His current focus includes cognitive modeling, mixed-initiative agent based systems, and simulation-based training applications. Dr. Diller holds a Ph.D. in Cognitive Science and Psychology and a M.S. in Computer Science from Indiana University.

RICHARD W. PEW is a Principle Scientist and BBN Technologies. He has 35 years experience in human factors, human performance and experimental psychology as they relate to systems design and development. Throughout his career he has been involved in the development and utilization of human performance models and in the conduct of experimental and field studies of human performance in applied settings.

KATHERINE GODFREY is a Statistician at BBN Technologies in Cambridge, MA, where she has been involved in projects in areas such as biomedicine, human factors, and speech analysis. She holds a Ph.D. in Statistics from Harvard University.

STEPHEN DEUTSCH is a Division Scientist at BBN Technologies in Cambridge, MA. His principal area of research is human performance modeling with related interests in agent-based systems and simulation.